

**RL-TR-97-70**  
**Final Technical Report**  
**August 1997**



# **DATA MANAGEMENT FOR AN INTEGRATED COMPUTATIONAL ENVIRONMENT**

**Capraro Technologies, Inc.**

**Gerard T. Capraro**

19971007 211

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

**DTIC QUALITY INSPECTED 4**

**Rome Laboratory**  
**Air Force Materiel Command**  
**Rome, New York**

Although this report references limited documents (\*), listed on pages, 25, 64, and 86, no limited information has been extracted.

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

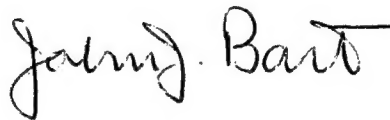
RL-TR-97-70 has been reviewed and is approved for publication.

APPROVED:



KENNETH R. SIARKIEWICZ  
Project Engineer

FOR THE DIRECTOR:



JOHN J. BART, Chief Scientist  
Electromagnetics & Reliability Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/ERST, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE <b>August 1997</b>	3. REPORT TYPE AND DATES COVERED <b>Final Mar 95 - Sep 96</b>		
4. TITLE AND SUBTITLE  <b>DATA MANAGEMENT FOR AN INTEGRATED COMPUTATIONAL ENVIRONMENT</b>		5. FUNDING NUMBERS  <b>C - F30602-95-C-0109 PE - 62702F PR - 2338 TA - 03 WU - PB</b>		
6. AUTHOR(S)  <b>Gerard T. Capraro</b>				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  <b>Capraro Technologies, Inc. 311 Turner Street, Suite 410 Utica, NY 13501-1731</b>		8. PERFORMING ORGANIZATION REPORT NUMBER  <b>N/A</b>		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  <b>Rome Laboratory/ERST 525 Brooks Road Rome, NY 13441-4505</b>		10. SPONSORING/MONITORING AGENCY REPORT NUMBER  <b>RL-TR-97-70</b>		
11. SUPPLEMENTARY NOTES  <b>Rome Laboratory Project Engineer: Kenneth R. Siarkiewicz/ERST/(315) 330-2465</b>				
12a. DISTRIBUTION AVAILABILITY STATEMENT  <b>Approved for public release; distribution unlimited.</b>		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) <b>In this report we are concerned with establishing a framework for an intelligent information processing paradigm for USAF weapon system development and maintenance. This involves the intelligent processing of data, knowledge, and information. Our goal is to design and develop a data management structure which will automate portions of this process. This approach will be prototyped and demonstrated using an actual system as related to reliability and electromagnetic sciences. The need for this work is evidenced by many factors. Technology is changing at an accelerated rate. The technology "know how" and expertise of many people are disappearing within corporations because of downsizing, acquisitions, and early retirements. Managing of information is important throughout the USAF. We must capture and harness information and obtain knowledge on a daily basis. If we lose information through the loss of people or poor documentation then we may be the victims of recreating the past with additional expense. We must capture the data, knowledge, and information and process it in an intelligent manner. The USAF is changing how it acquires, builds and maintains its weapon systems. It is not immune to information, inaccurate data, and "cloudy" knowledge. This report and the work that flows from it will result in a better and more efficient way to survive in this information driven world.</b>				
14. SUBJECT TERMS  <b>Electromagnetics, Artificial Intelligence, Database Management, Knowledge bases, Heterogeneous Databases, Intelligent Agents</b>		15. NUMBER OF PAGES  <b>96</b>		
17. SECURITY CLASSIFICATION OF REPORT  <b>UNCLASSIFIED</b>		18. SECURITY CLASSIFICATION OF THIS PAGE  <b>UNCLASSIFIED</b>		16. PRICE CODE
19. SECURITY CLASSIFICATION OF ABSTRACT  <b>UNCLASSIFIED</b>		20. LIMITATION OF ABSTRACT  <b>UL</b>		

## TABLE OF CONTENTS

	Page
Abstract	1
1. Introduction	2
2. Overview	3
3. Research and Engineering Framework (REF)	9
4. An Integrating REF Structure	13
5. A Functional DBMS Integration Process	16
6. The Future	19
7. Conclusions	24
References	25
Appendix A Some DBMS Tools	27
Appendix B ICE Database System Requirements	31
Appendix C First Interim Report	34
Abstract	34
1. Introduction	35
2. Overview	36
3. Rome Laboratory Related Efforts	39
4. Microwave and Millimeter-Wave Advanced Computational Environment (MMACE)	42
5. Integrating Databases	47
6. Conclusions and Recommendations	47
Appendix C1 - Integrating Databases	50
References	64
Appendix D Second Interim Report	66
Abstract	66

1.	Introduction	67
2.	Overview	68
3.	Research and Engineering Framework (REF)	74
4.	An Integrating REF Structure	78
5.	Integration of Multiple Domain-Specific Relational DBMSs	84
6.	Summary	85
	References	86

## List of Figures

	Page
Figure 1. Conceptual View of ICE	4
Figure 2. Integrated Heterogeneous Databases	6
Figure 3. EMC Life Cycle Data Requirements	8
Figure 4. REF Elements	9
Figure 5. REF Database Population Sequence	10
Figure 6. REF Database and Code Interface	11
Figure 7. Re-Hosting REF Elements on a DBMS	13
Figure 8. Integrating Two Different Database Domains	14
Figure 9. An Integration Process	16
Figure 10. Performing a Global Query	18
Figure 11. A Integrated Knowledge Base System	21
Figure 12. A Data, Knowledge, and Intelligent System	23
Figure B-1. Integrated Heterogeneous Databases	31
Figure C1. Conceptual View of ICE	37
Figure C2. EMC Life Cycle Data Requirements	38
Figure C3. Integrated Heterogeneous Databases	39
Figure C4. REF Elements for MMACE	43
Figure C5. REF Database Population Sequence	44
Figure C6. REF Database and Code Interface	45
Figure C7. REF Database and Code Relationships	46
Figure C8. A REF/DBMS-based ICE Approach	48
Figure C9. A TRADE-based ICE Approach	48
Figure C10. Three Level View Model	50
Figure C11. An E-R Diagram	52
Figure C12. An Object Diagram	53

Figure C13.	Relational Tables	54
Figure C14.	A Network Diagram	55
Figure C15.	A Hierarchical Diagram	56
Figure C16.	DBMS System Structure	57
Figure C17.	A Data Warehouse Architecture	63
Figure D1.	Conceptual View of ICE	69
Figure D2.	Integrated Heterogeneous Databases	71
Figure D3.	EMC Life Cycle Data Requirements	73
Figure D4.	REF Elements for MMACE	74
Figure D5.	REF Database Population Sequence	75
Figure D6.	REF Database and Code Relationships	76
Figure D7.	Re-Hosting REF Elements on a DBMS	77
Figure D8.	Building Two of the Integrated Databases	79
Figure D9.	The REF Structure and Vacuum Electronics Integrated Databases	80
Figure D10.	EMC Code Relationships	81
Figure D11.	The REF Structure and EMC Integrated Databases	82
Figure D12.	The REF Structure and the Integration of Two Different Database Domains	84

## **Abstract**

This is the final report documenting the results obtained in performing a study to design a top-level data management scheme for an integrated computational environment (ICE). A description of ICE is provided along with a design of how the Microwave/Millimeter-wave Advanced Computational Environment (MMACE) program's Research and Engineering Framework (REF) can be used as a foundation for building ICE. A description is provided that demonstrates the integration of heterogeneous databases within the same domain and from multiple domains of interest (i.e. vacuum electronics industry and electromagnetic compatibility). The ICE design is extended providing the foundation for knowledge bases and intelligent agents to access heterogeneous data in a seamless and consistent manner.



## 1. Introduction

The Integrated Computational Environment (ICE) is an approach for designing and modeling components, boards, boxes, line replaceable units (LRU), subsystems and systems for the USAF. The Department of Defense (DoD) is slowly moving towards the use of modeling and simulation techniques for fulfilling part of the functions that have been performed by military specifications, and testing. The old approach was based upon the premise that if each component met the military's specifications then when the full system was integrated it would meet the military performance and environmental conditions. This approach in many cases led to over-designed components and increased costs because the commercial market did not require these designs and could not afford the extra quality. The new trend of using commercial parts, when shown feasible through analysis, modeling and simulation, should bring down the cost of military systems by making use of less costly commercial off-the-shelf (COTS) hardware and software.

To implement the ICE approach within the DoD is in itself a challenge. The challenge lies on many fronts, from acquisition policies, to testing, to maintenance, to rights of ownership of data. This particular contractual effort is concerned with the challenge of designing the integration of the different modeling and simulation tools such that Concurrent Engineering (CE) can be performed using these tools and thereby reducing the cost of procuring military systems.

This is the third and final report within this contractual effort and covers the results of the third task and reviews the previous two tasks which are documented in Appendices C and D respectively. The third task is to:

"Develop a design and plan for the building of an ICE. This shall include a description of the problem, a requirements definition, a "high level" proposed solution described in functional "block" diagram form, descriptions of each block with estimates of resources to complete, and a flow diagram over time illustrating a plan to build, integrate, demonstrate and validate the ICE."

The above task was modified because of the changes that occurred from the time the statement of work was completed and the onset of this effort. A plan to demonstrate and validate ICE was replaced with adding knowledge and intelligent processing to the proposed architecture. The third task looked at the different related technologies to provide the integrated data at different levels of management for knowledge and intelligent processing. This report documents how the Research and Engineering Framework (REF) being developed as part of the MMACE program can be used as a foundation

for building ICE and provides an overview of the tools available for integrating data, knowledge, and intelligence.

The first report provided an overview of the ICE and the motivation for its existence. It also provided a description of those projects within Rome Laboratory that are directly related to ICE, a description of the REF portion of the MMACE program and a short tutorial on Database Management Systems (DBMS) and the integration of heterogeneous databases.

The second report provided a more in-depth description of the ICE concept. This was followed by a discussion of the REF and how it can be enhanced by hosting some of its elements on a Relational Database Management System (RDBMS). A description of how the REF structure can be used to integrate heterogeneous databases within a defined domain of interest (e.g. the vacuum electronics industry, the Electromagnetic Compatibility technology area) was provided along with how the REF architecture provides the basis for building an integration of heterogeneous databases from multiple domains. The first and second reports are contained in Appendices C and D, respectively, and are referred to periodically throughout this report.

This final report provides a brief overview of the ICE concept and our findings to date. The next section provides an overview of ICE. This is followed by two sections related to the REF and how it can form the basis for integrating heterogeneous databases. Section 5 provides a functional integration plan for developing ICE. Section 6 provides a next generation design for ICE and how global users can efficiently use the integrated databases for obtaining knowledge and information in a "point and click" and timely manner. The report concludes with a section which provides some of the benefits that can be obtained by implementing ICE.

## **2. Overview**

The Rome Laboratory is developing technology to help design and build new or improved weapon systems with the highest reliability, compatibility, and maintainability while using commercial components and minimizing costs. The military acquisition process for purchasing systems with military specifications and standards will be changed over the next few years. Methods to integrate commercial components into military systems will rely heavily on computer modeling and simulation as opposed to standards and testing.

There are, however, several sources of inefficiencies and inaccuracies in the current use of modeling and simulation for the acquisition of DoD systems. The DoD simulation and modeling tools/codes available for system development and deployment were built by many different technologists/disciplines, with each code

and its data related to its own area. In addition, the people concerned about reliability, compatibility, and maintainability normally are not involved early in the design process nor in the deployment modeling process. When they are involved, they are sometimes evaluating data and designs that have been changed or they are involved after the system is deployed and is not functioning as designed or expected.

An approach to minimize these problems and inefficiencies is to define and implement an Integrated Computational Environment (ICE). This computational ability must provide a consistent and obtainable database, describing an overall system, its components, and its environment, and must provide the capability of integrating *government and commercial* data, modeling, and simulation tools. The ICE should be relatively transparent to the current tools and methods that are in practice. However, it should provide the compatible framework for integrating the different databases, tools, models, and simulation packages, such that well-defined interfaces can be established and controlled for a more efficient, timely, and accurate exchange of data. A conceptual vision of ICE is shown in Figure 1.

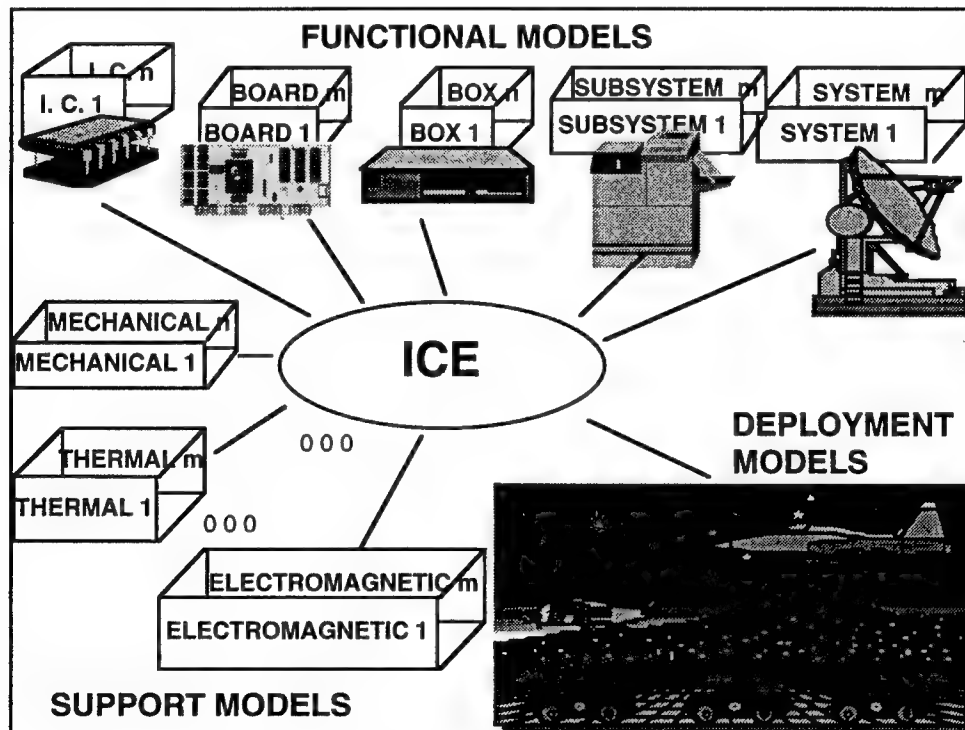


Figure 1. Conceptual View of ICE

ICE is a structure for integrating functional models, support models, and theater-level deployment models. Functional models are those models used to develop the components of a system to

meet a system's primary performance requirements. The throughput of a computer, the sensitivity level of a communications receiver, and the radiated power of a radar are examples of system components' primary performance requirements. The support models are those models that are concerned with a component meeting a system's secondary set of requirements. These are usually related to environmental concerns such as mechanical, thermal, and electromagnetic. Theater-level deployment models are related to that process of evaluating new or unavailable components to determine their performance in actual and varied deployment environments. These models may be strictly digital simulations or they may be composed of a mixture of actual components, digital simulation models, and components which emulate other components. With the proliferation of computers within most military systems and the reduced DoD budget, it is becoming more common for the military to exercise theater-level simulation and/or emulation models to evaluate new or proposed military systems rather than building a prototype system.

The development and deployment process of a new system, e.g., radar, aircraft, or missile, is very complex and involves many people with varied capabilities and objectives. It usually requires a prime contractor and several subcontractors with many people at different locations. These people can be divided into three basic groups based upon their interests. Group 1 consists of those people interested in building a system's components, e.g., high power tubes, processors, amplifiers, sensors, power supplies. An example may be a sub-contractor or a component provider or supplier. Group 2 consists of those people interested by technology or support function, e.g., circuit design people, thermal, electromagnetic, structural, signal processing, communications, radar, contracts, legal, accounting. Group 3 are those people interested in the system-level effects of integrating a system within the deployment environment e.g., system simulations, system emulations, battlefield simulations/emulations. These three groups can be partitioned further by the data required of the computer applications or codes used in an individual's job, e.g., the computational electromagnetic (CEM) area is composed of many codes some of which treat electrically small structures, while others model electrically large structures.

Consider the potential benefits gained if the data requirements of these different groups were consistent, computerized, secure, and instantly accessible anywhere throughout the world. Connection to a global database from any terminal with a modem would allow for the retrieval of the most detailed data instantly. This capability would reduce the cost and compress the schedule of system development, deployment, and maintenance throughout a system's cycle, while enhancing performance and safety. *The computer technology to accomplish this is here today; but the*

methods and tools for integrating the data among the three different groups is not in place. As an example consider Figure 2.

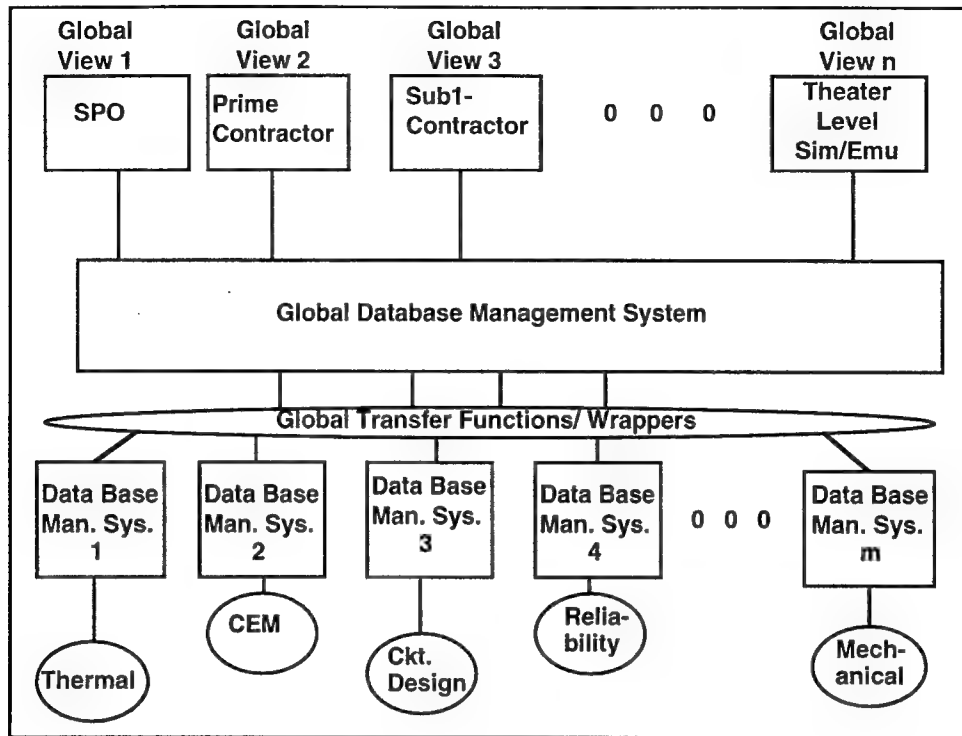


Figure 2. Integrated Heterogeneous Databases

Figure 2 illustrates an approach for integrating a collection of heterogeneous databases from the bottom up. The bottom portion of the figure depicts each set of users partitioned by technology (i.e., Group 2). Each user within a technology would have a consistent database that represents any component of interest across all of the codes that are used in that technology over the life of the component. The different databases (thermal, CEM, design, etc.) would be integrated into another consistent database by the Global Database Management System (GDBMS). This allows all users access to the total database whether they are a technology modeler (Group 2), a sub-contractor (Group 1), or a Government agent assessing new technologies in a simulated battlefield environment (Group 3). Access to the data within the GDBMS can be obtained within any group given the need to know. The data can be stored at one location centrally located or across a distributed network of computers. Data can be obtained in "real-time" for analysis, meetings, inquiries, and reporting at any location with a computer and a modem.

To obtain a consistent set of data that is available to many throughout the development and deployment of a weapon system, we must begin building a structure based upon existing data that are

already being gathered by the respective groups. (See the bottom portion of Figure 2.) In modern-day systems the digitization of data usually takes place when people begin to design the system's components. They primarily use computer codes accepted by the community and/or company proprietary codes. However, it is the data, not the codes, that drive the requirements for an architecture like that shown in Figure 2. In many organizations the individual users are using their own codes and are not sharing data via a database management system. It is this level of the architecture that must be integrated first. To start the process by defining the data requirements from the users at the top level of the architecture (i.e., the global viewers at the top portion of Figure 2) would be too costly. More importantly, this would disrupt the current process.

As an illustration of the data involved, consider the electromagnetic compatibility (EMC) community. Figure 3 illustrates the data required by the EMC community for different components and at different stages of a component's development and deployment. The EMC community uses a subset of the codes within the CEM area. The data required by most of the different users within the Groups are dependent upon their codes, the component of interest (e.g., radar, integrated circuit), the acquisition stage, and the deployment environment of the component.

	INTEG. CKT.	BOARD	BOX	LRU	SUB- SYSTEM	SYSTEM	THEATER LEVEL
CONCEPT FORMULATI ON	NULL	NULL	NULL	NULL	Antenna & Wire Char.	Freq. Allo- cation	EM Environ- mental Survey
REQUIREME NTS DEFINITIO N	NULL	NULL	G/B/S*	G/B/S	Antenna & Wire Char.	Same	Power Cull & Propaga- tion Studies
SPECIFICA TIONS DEVELOPME NT	NULL	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation	Antenna to Antenna & Environ- mental Coupling Analysis
PRELIMINA RY DESIGN REVIEW (PDR)	Ground- ing, Line Widths, SPICE /Ckt Codes	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation	Antenna to Antenna & Environ- mental Coupling Analysis
CRITICAL DESIGN REVIEW (CDR)	SPICE /Ckt Codes	Wire Codes, Field Codes, & SPICE/ Ckt Codes	IEMCAP, Field Codes, & SPICE/ Ckt Codes	IEMCAP, Field Codes, & SPICE/ Ckt Codes	IEMCAP & Field Codes	IEMCAP, Field Codes, & Freq. Assign- ment	Inter- platform EM Dynamic Simula- tion/Emu- lation
INSTALLAT ION AND TESTING	NULL	Wire Codes, Field Codes, & SPICE/ Ckt Codes	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing & Monitor- ing
MAINTENAN CE	NULL	G/B/S/R* Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	Same + Testing & Monitor- ing

\*G/B/S/R Implies Grounding/Bonding/Shielding/Radiation

Figure 3. EMC Life Cycle Data Requirements

There is one of these matrices for each technology discipline and for each management function (e.g., accounting, contracts, legal). Data integration must begin within each of the technologies. For the most part, the analysts and engineers within each technology presently use different codes and are not integrated nor share their respective data in any computerized efficient form.

The building of this architecture is based upon the sharing of data generated through the use of computerized tools. Note that once the data are integrated and maintained as shown in Figure 2,

it can provide the basic data and/or "facts" for knowledge based and intelligent systems. This approach is expanded upon in Section 6.

The building of the architecture shown in Figure 2 begins by integrating data at the lowest of levels. How does one integrate data required by heterogeneous codes within the same technology and across multiple engineering disciplines? This area is being addressed in the Tri-Service Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research and Engineering Framework (REF) development program and is discussed in the next section.

### 3. Research and Engineering Framework (REF)

The MMACE program is a Tri-Service and NASA initiative to improve the power tube design process. It is composed of two portions. One portion is composed of the vacuum electronics codes and tools that are used to perform the design and analysis of power tubes. The second portion is the Research and Engineering Framework (REF) which contains the programming interfaces, standards, and utilities to aid in the integration of the codes and tools. A diagram of the REF is shown in Figure 4, and the reader is directed to references (1-3) for an overview.

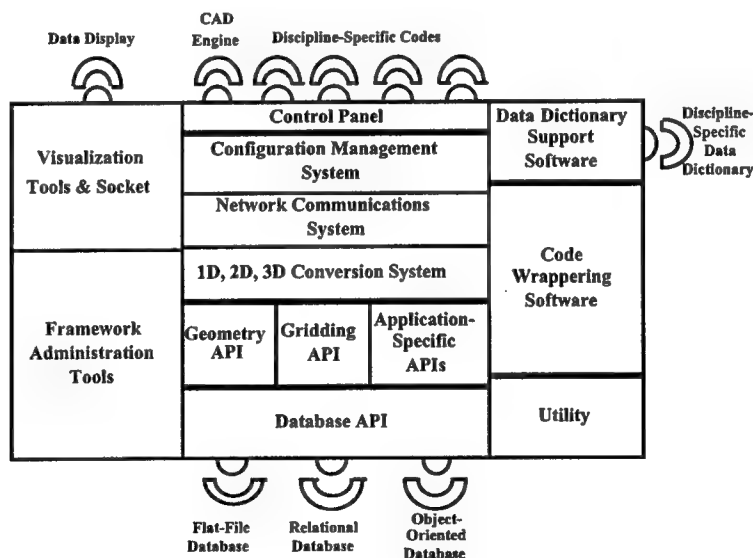


Figure 4. REF Elements

The REF is being developed for a well-defined set of requirements, for a small industry, and with a limited budget. The following is a brief description of the REF. Figure 5 shows an overview of the



REF's ability to interface with a user and the population sequence of the integrated database. The vacuum electronics industry has a finite set of analysis tools which apply to the different stages or elements of a microwave/millimeter-wave tube. Each of these tools requires Initial Graphics Exchange Specification (IGES) input files, both geometry or parametric data, for it to operate. The user can describe the portion of the tube using a Computer Aided Design (CAD) package that generates IGES files. Because different CAD packages generate different IGES-compatible files for the same design, the REF developers chose a NASA-defined subset of the standard IGES file description to store in the database. This required them to develop an IGES translator that can read the output of a commercial CAD package and convert it to a standard format or specification such that data incompatibilities would not exist between different CAD packages operating on the same computer or entity. They have written these software tools to operate with the AutoCad and ProEngineer CAD programs.

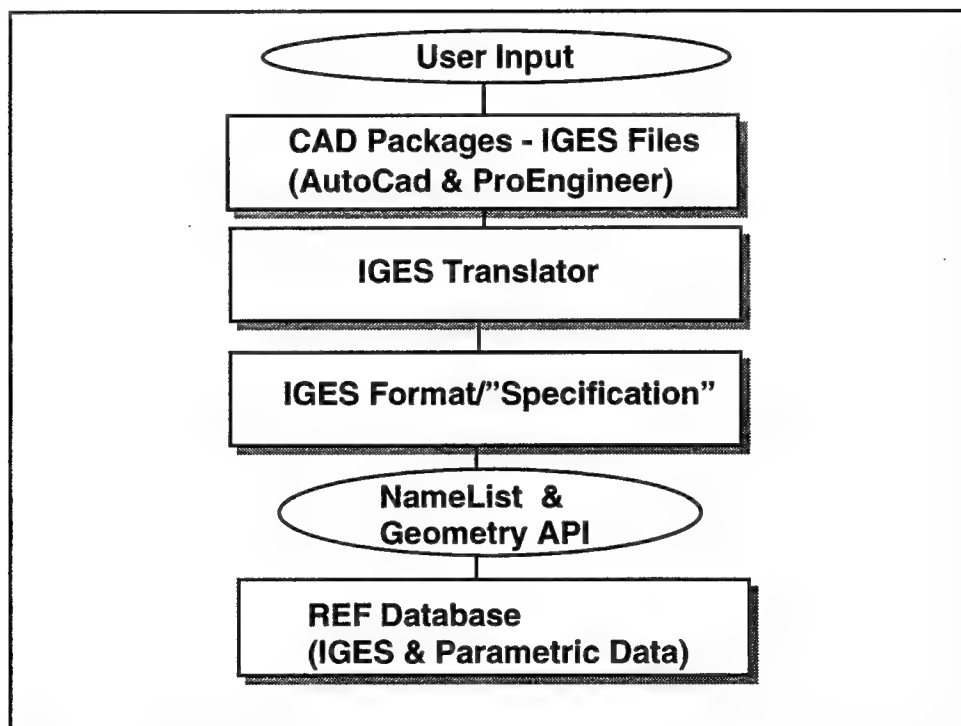


Figure 5. REF Database Population Sequence

The integration of the different data required by the different codes is performed mainly through two approaches. The geometry data parameters are controlled through the use of the CAD packages and their IGES file formats. The naming conventions and/or parametric data are controlled by the tube industry through consensus. That is, each code has access to and must stay

compliant with a fixed set of parameters, units, names, etc. This forces the community to have a homogeneous database with few parameters that are code-dependent, i.e., lie outside their common intersection. Figure 6 depicts a subset of the codes, in which each set in the Venn Diagram represents a tube code and its input parameters. Few attributes (or input parameters) are code-dependent and not shared. The four codes identified are those that have unique attributes to describe the model. The Shared Data, the center set or major intersection, is accessed by eight or more codes.

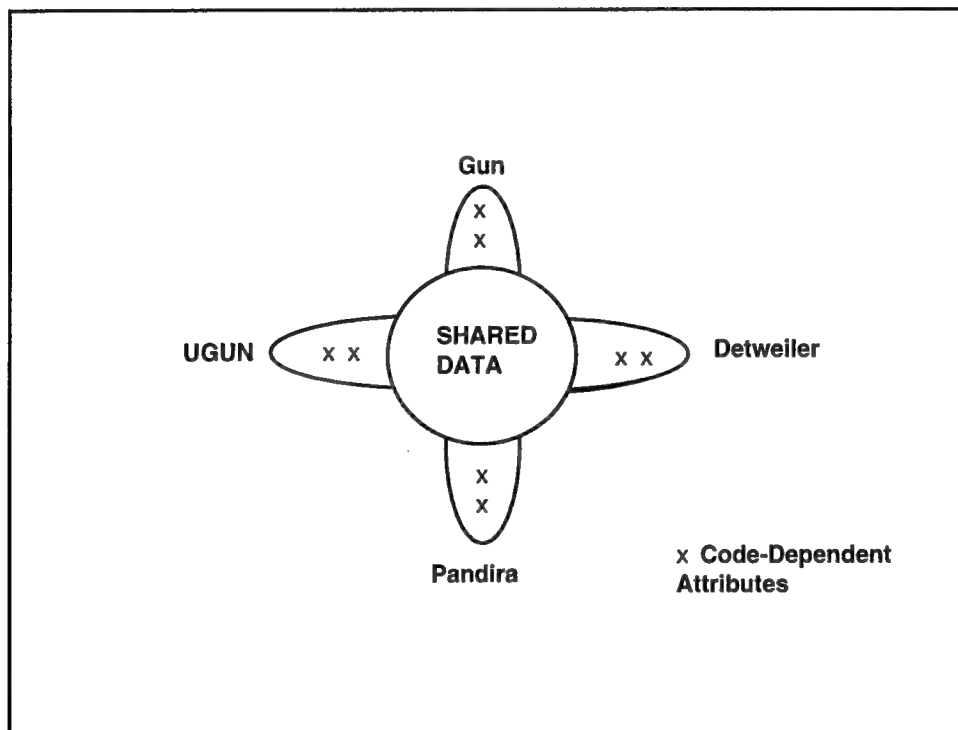


Figure 6. REF Database and Code Relationships

The REF also has a Data Dictionary (DD) which maintains a list of the attributes within the database. A DD within a DBMS stores meta data and authorization information, such as key constraints and user privileges, and is the direct interface to the database. (Meta data are those data about the data, e.g., an attribute's name, field type, and size of the field.) The DD within the REF only performs a bookkeeping function that allows one to query which attributes are in the database, but it is not capable of searching the database for the values of these attributes. The DD is as up-to-date as the industry manually maintains its contents. This is an important issue since adding new data to the database is easy. However, changes to the database affect the DD and all wrappers interfacing codes to the database. *The industry must manually update the wrappers and the DD when one adds, deletes, or*

changes the database schema or design. This manual process could be simplified if the DD and the database were implemented with a DBMS. This would provide data independence from the application tools and the wrappers and would minimize the cost for maintaining the system. Data independence allows one to change the database design and contents while minimizing the effect to the application tools and wrappers.

Integrating a DBMS within the REF will enhance its capabilities, reduce its maintenance cost, and increase its robustness and growth potential. Areas within the REF that can take advantage of a full DBMS are shown in Figure 7, which contains the same functional blocks as the conceptual diagram shown in Figure 4. The shaded portions indicate those areas where modifications to the REF can be performed. A portion of this integration process will be re-hosting pieces of REF on a DBMS and using commercial software tools to help integrate databases. The Control Panel can be updated allowing the user access to forms for user-friendly building of queries and reports from the DBMS. These forms would add to the current capability for executing jobs within the REF. The Data Dictionary Support Software and Discipline Specific Data Dictionary functions can utilize the DBMS's imbedded data dictionary capability, e.g., its software algorithms for defining data, setting priorities, defining key words, access control, and integrating the different data definitions within domains and between domains. Database APIs are those tools that allow for report generation and query support for the casual user and for the domain specific database administrator. The Framework Administration Tools help in maintaining data integrity and concurrent engineering functions required by the different domains. Some tools within the chosen DBMS can replace current REF tools and/or work in concert with them and add increased functionality.

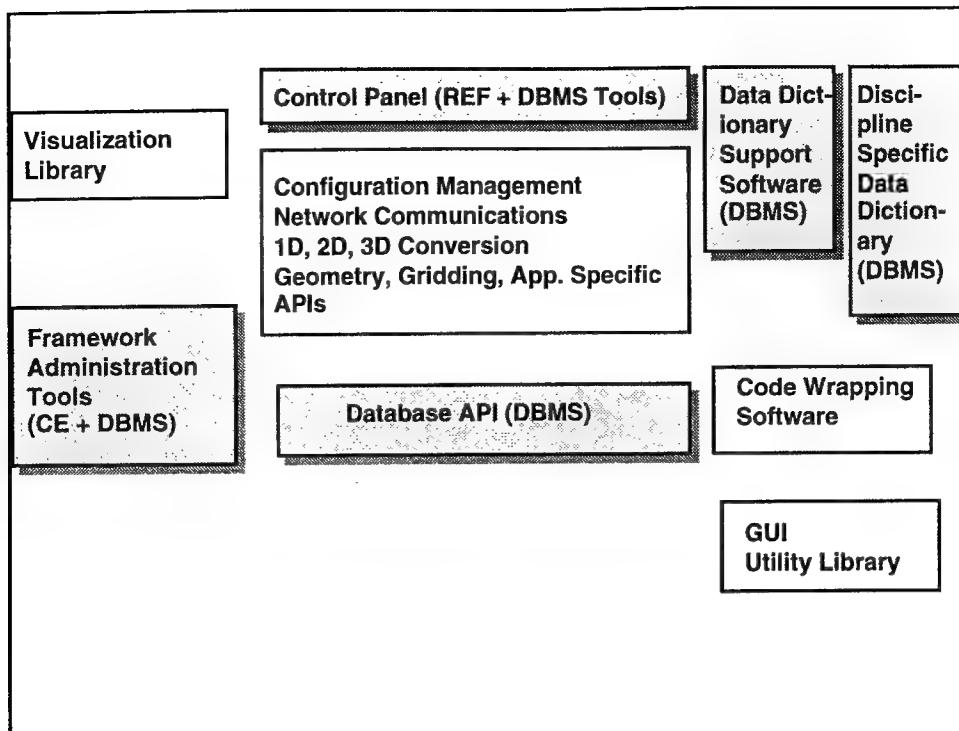


Figure 7. Re-Hosting REF Elements on a DBMS

#### 4. An Integrating REF Structure

The previous section provided an overview and proposed a DBMS extension to the REF software architecture. This extended REF can be the foundation for integrating data from other domains. The process is documented in Appendix D; where descriptions of how to integrate the data from multiple tools within the same domain and how to integrate the data between multiple domains is presented. This approach allows for the building of a consistent Data Dictionary and Data Directory (DD/DD) for the Global DBMS, which does not contain domain data, but the data dictionary and directory data (i.e. meta data). It allows users access to the DBMSs similar to the domain users when forming queries to its databases. This DBMS interface to the data assures data consistency and integrity. The maintenance of the DD/DD must be in cooperation with the separate DBMSs. That is, if the individual DBMSs make changes to their DD/DD then the Global DBMS must also be changed. Otherwise its application software and accesses to the DBMSs will be in error or will not execute.

A working group with representatives from each of the individual DBMSs is one way of cooperating in building and maintaining consistency of the integrated DD/DD. Another way is to assign a committee to oversee and approve changes to the individual DBMSs before they are implemented. The best implementation method is

organization dependent. The common factor for success is to realize that good communication and cooperation are necessary.

In the second interim report (Appendix D) a description of how two different domains (Vacuum Electronics and EMC) can be integrated at the data level is provided. (See Section 5 in Appendix D.) The following figure provides a description of that integration.

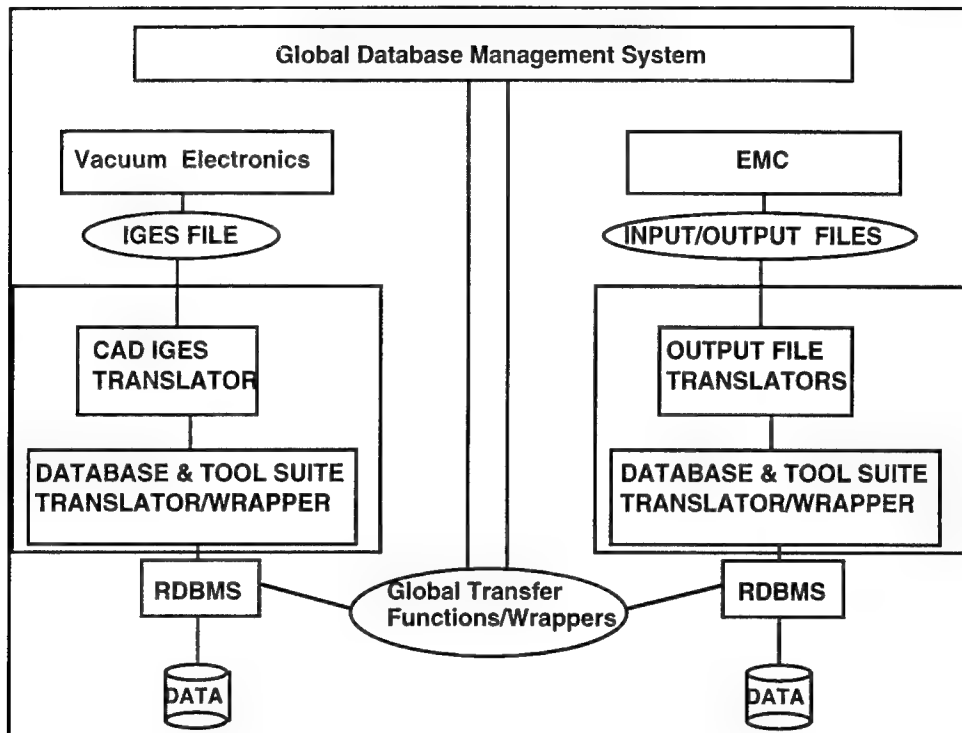


Figure 8. Integrating Two Different Database Domains

There are commercial tools to aid users in performing database integration both within a domain and between domains. The number of these tools has multiplied over the past two years because of the increased attention given to Data Warehousing. A Data Warehouse is a system whose components include people, architecture, process, procedures, software, and hardware. An objective of a Data Warehouse is to improve the quality and accuracy of information. To do so usually requires efficient access to integrated data from multiple, heterogeneous, autonomous, and distributed information/data sources (e.g. databases).

To meet the above objective a Data Warehouse makes use of old and current data to answer queries, determine data trends, compute statistical parameters, and make projections based upon old and current data. In order to meet these needs of obtaining information from "Gigabytes" of data the concept of Data

Warehousing was formed. The core of this architecture/system is integrating databases that may be located throughout the organization on different types of computers, on different DBMSs, and contained in many different legacy databases. To meet these needs many different tools are being sold to help organizations integrate databases. Many organizations today, including the Government, are "data rich and information poor". Data Warehousing is an attempt to increase the information level of corporate America.

The architecture shown above in Figure 2 can be viewed as a Data Warehouse, except it is more. The users at the upper level are interested in the same questions and queries that one would want from a Data Warehouse but the users at the bottom of the architecture are more interested in the data and their values as they are changed in real time. (This is not a requirement for a Data Warehouse system.) This requires very tight controls over the maintenance of the databases and configuration control of the data values and their respective schemas.

However, the database tools for building a Data Warehouse are very applicable for building the architecture shown in Figure 2. Some of these tools help build a data dictionary, some help one reengineer a current database, some allow and aid the integration of databases resident on multiple DBMSs. For instance (4), Logic Works Inc.'s ERwin/ERX Family of tools provides: "A database design tool for client/server development that lets a user point and click to design a graphical entity-relationship (ER) model for the business rules governing the data in their applications. Features forward and reverse engineering, and gives users a direct connection to their system catalog, creating a data model straight from their database tables. Changes to the data model can be forward-engineered to update the current database, or used to create a new database in more than 20 supported DBMSs. Tables, indexes, referential integrity (primary key and foreign key), defaults, domain/column constraints, and thousands of lines of stored procedure and trigger code, are all generated automatically, providing a solid foundation for new development. Also available in versions that support Visual Basic, PowerBuilder, or SQL Windows, synchronizing application development with the database design. Extended attributes can be captured and defined from within the ERwin data model itself and passed through a bi-directional link, providing the client side with a blueprint consistent with the server. Ready-to-run, data-aware Visual Basic and SQL Windows Forms and PowerBuilder DataWindows can be generated directly from the ERwin database design." (<http://www.logicworks.com>)

Another tool discussed in (4) is developed by Embarcadero Technologies, Inc. is ER/1: "An advanced entity-relationship modeling tool. Inheritance engine ensures the proper migration

and unification of foreign keys between entities, building referential integrity into ER diagrams automatically. Integration with major database platforms includes tables, table constraints, primary and foreign keys, indexes, triggers to maintain referential integrity, stored procedures to perform data manipulation, and shadow views. Can also be used to document existing databases. Can X-ray the structure of a database and reverse-engineer its schema into an ER diagram. Compatible with Oracle7, Sybase System 10, Microsoft SQL Server, Watcom SQL, Informix, DB2/2 and SQLBase." (<http://www.embarcadero.com>)

There are numerous tools like the two presented. A list of some of these tools found on the World Wide Web (WWW) is presented in Appendix A. A complete list is not intended nor are we proposing any one tool over another. To build the above designed architecture can require one or more tools depending upon the DBMSs that are to be integrated and the DBMS chosen to host the meta data within the Global DBMS.

## 5. A Functional DBMS Integration Process

A description of how to integrate the data and databases within and between technology domains has been presented. A plan for implementing this approach is presented in this section. A high level or abstract view is presented in Figure 9.

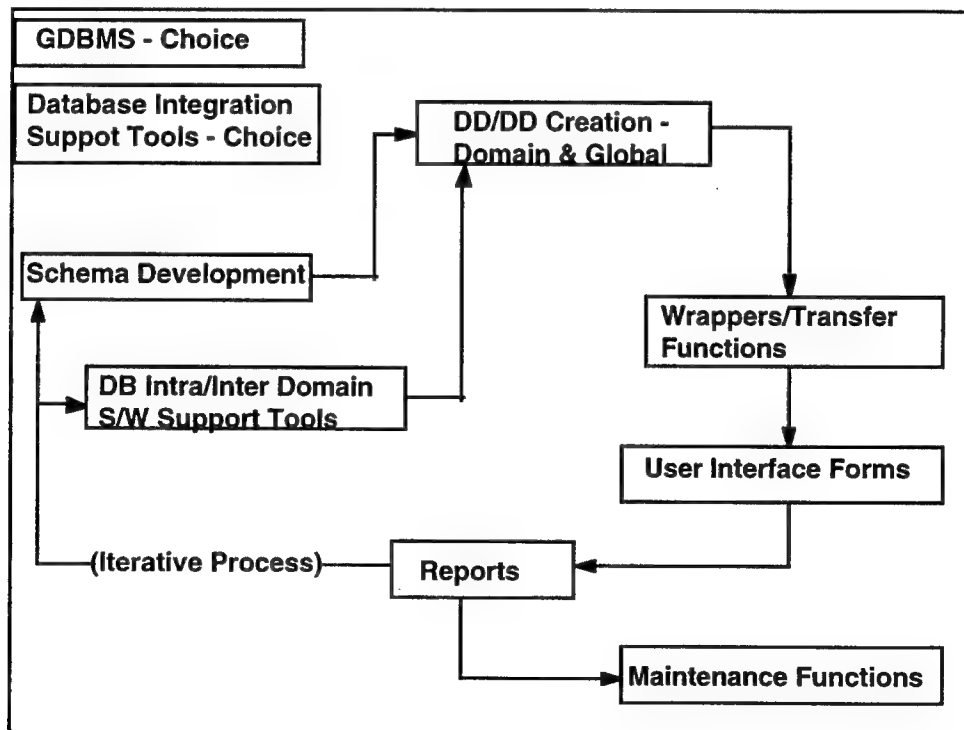


Figure 9. An Integration Process

There are two basic studies that must be performed before any major investment is expended in integrating any of the DBMSs. First a DBMS must be chosen to perform as the GDBMS. This may depend on many factors for example cost, computer hardware, and support tools. To focus this process a set of requirements is provided in Appendix B. A preliminary survey of four well-known commercially available high-end DBMSs was performed and they all met the requirements. Performing this function is estimated at two person weeks by a senior and knowledgeable DBMS expert.

In concert with choosing the GDBMS one or more support tools for performing the database integration needs to be chosen. A brief description of these tools and a list of some of them are provided in Appendix A. These two tasks performed together will provide the best tool set for building a software development environment, since not all support tools work with all GDBMSs. Performing this function is also estimated at two person weeks by a seasoned programmer with DBMS experience.

In Appendices C and D and in this report we have discussed the integration of data and databases from intra-domains and inter-domains. The effort required to perform this exercise is difficult to estimate. It is assumed that for the following tasks that the person has at least fifteen years experience in databases and in software development with at least a masters of science degree in computer science or computer engineering. First one must develop a relational DBMS schema of the database for each CAD or analysis tool. Then it must be interfaced with the existent DD/DD. Wrappers/functions must be written for those attributes whose values are homonyms and/or synonyms of attributes currently within the Global DD/DD and/or their meta data description is not identical to the same attribute in the Global DD/DD. The amount of effort to perform this task can be as low as one person month to multiple person-months per database.

The development of the user input forms for creating queries to the meta data, query generation to the DBMSs, report generation functions, schema modification functions, interface functions, support for the development of the transfer functions and wrappers, and overall maintenance of the DD/DD must also be performed. These functions would be developed by a programming staff and a Database Administrator (DBA). The amount of effort required is directly dependent upon the complexity of the different databases, their number, and the frequency, complexity, and amount of different applications and users interfacing to the GDBMS.

To visualize how this architecture may be implemented, consider an integrated system with multiple DBMSs. Assume that the schema for the GDBMS has been created and it consists of numerous relations



as shown in Figure 10. The GDBMS contains and maintains the DD/DD for the integrated databases and the data are maintained and stored within the individual domain DBMSs (e.g., thermal and CEM). If a query or report is submitted to the GDBMS that involves relationship AA, the system first recognizes through its own DD/DD that AA contains attributes whose values are obtained by implementing transfer functions A1, A2, and A3. These transfer functions could be implemented as Open Database Connectivity (ODBC) queries to the proper domain DBMSs (e.g., QA1). These transfer functions would know about homonyms and synonyms via the global DD/DD. The return of the queries from the domain DBMSs would then be exercised by mapping functions which would map their values to the proper formats for their respective global attributes contained in AA. These functions would know about the differences in integer, floating point, date types, binary variables, etc. The results would be stored in temporary tables. These tables would then go through any projections and/or joins in order to create the resultant occurrences for populating relation AA. Then the original query or report written for the GDBMS would operate upon this table and provide the result to the global user (e.g.,  $F1(F(A1, A2, A3))$ ). The table containing the occurrences of AA could be provided to the user or saved within the GDBMS depending upon performance, cost, and maintenance issues, or it could be deleted upon termination of the global user's connection to the GDBMS.

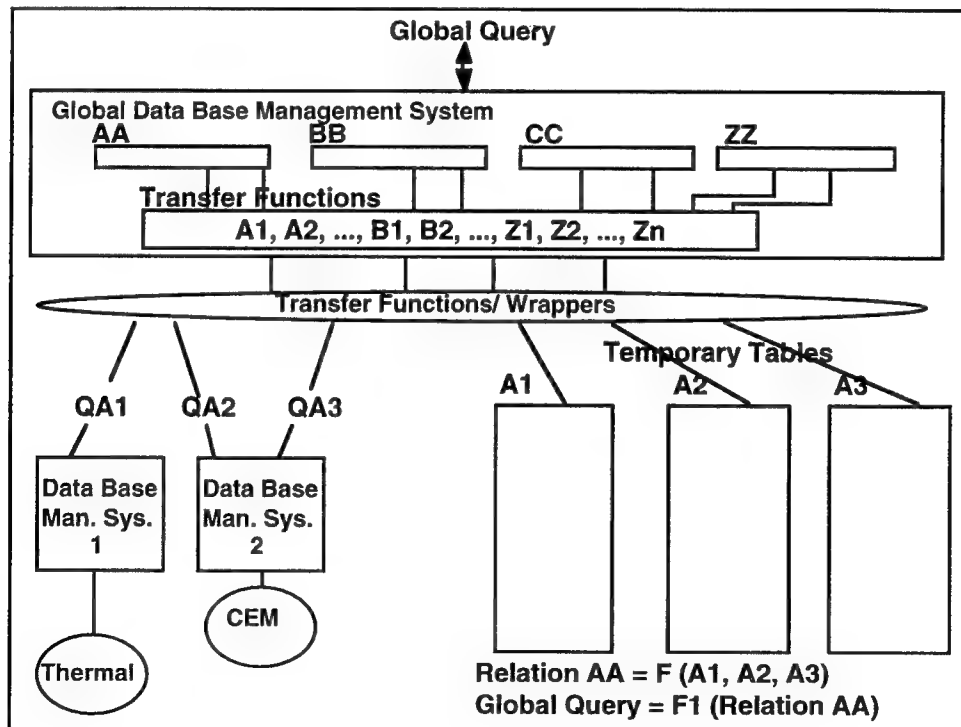


Figure 10. Performing a Global Query

There are many ways to create and maintain the global database. Some configurations would let the data be stored only within the domain databases, as mentioned earlier, and only access them to answer queries and delete their contents upon completion. This approach insures data consistency because there is only one global database. Another approach would be to create the global database and store it within the GDBMS and then populate any changes made within the domain databases periodically. This can be performed in "real time" or every day for instance, depending upon the volatility of the domain databases.

This integrated global DBMS system will allow for the creation of queries that span multiple codes within one domain and/or queries that span multiple databases across domains. For example, a system engineer may want to know all the current locations of antennas on an aircraft. Based upon the results of the query a second query may be to provide any data related to antenna patterns of one or more of these antennas; whether the patterns were simulated, measured in a chamber, free space, or on a mock-up. What is the status of the antenna's development; i.e., has it passed preliminary design review yet? These kinds of queries would require accessing different DBMSs. The user only needs to communicate to one DBMS, learn one set of protocols, and retrieve consistent accurate data, in a timely manner.

## **6. The Future**

The previous sections have outlined a design for integrating numerous disparate databases so that they can be viewed as one homogeneous set of data within a consistent database. However, the interface to all this wealth of data is through a DBMS. This is great if the user is a programmer, engineer, or database person or one has such a person always available at a moment's notice. They would be needed to query the system and provide a manager with the results of ad hoc queries at a moment's notice. In addition, as databases are added to the structure, the knowledge of what queries can be asked will be dynamic. The user's query capability will change as new data are added to individual DBMSs at the domain levels. Methods would have to be developed of letting the global users know how and when new information can be obtained and when different events have occurred. The system is not passive; it needs to let its users know when and how its contents are changing.

The paradigm of a database being relatively static and the only changes are to the values of attributes is not the case within this architecture. The data are changing but so are the attributes. This integrated database approach provides a wealth of data that is changing in real time both in structure and in content. It is somewhat analogous to the internet. Every time

you search the net you retrieve or find additional data for the same or similar queries. This is because new addresses for data have been added to search engines, new web pages have been added, the contents of web pages have been changed, a different search engine is used, different key words are used, etc. The reasons are many. However, the internet represents uncontrolled growth. In this relatively closed system, it would be harmful to allow uncontrolled growth. Enforcing controlled growth at the GDBMS and domain database levels must occur for data consistency, accuracy, security, and maintenance.

There are two issues that must to be addressed. How to bring this tremendous amount of potential data to the global users (Generals, vice presidents, general managers, SPO chiefs, comptrollers, program managers, chief engineers, etc.) in a "point and click" and timely manner? How to alert and efficiently integrate new changes to the GDBMS for the global users?

These issues can be addressed in different ways. A proposed approach is to let the general user access the data via database queries, a knowledge base goal achiever, and/or an information or intelligent agent mode. See (5) regarding a good overview of intelligent Executive Information Systems (EIS). Depending upon the global user and his or her requests, different modes of accessing the heterogeneous databases may be appropriate. For standard reports for example, related to schedule updates, expenditures to date, milestone slippages, travel budget projections, transaction reports written against the GDBMS can provide the information to the global users. Random or ad hoc query capability can be provided to the user with a point and click interface for most types of queries. This will allow a global user to formulate and execute his or her own queries to the GDBMS. These interfaces to the system can also be performed using voice input, keyboard, and/or mouse interactions.

Along with this capability it is possible to provide a knowledge base system "on top" of the heterogeneous databases. The GDBMS would provide the fact base for the "rules" contained within the knowledge base. Different knowledge based systems can be created to support the different needs of the different global users. A knowledge base system could be used to extract data from the GDBMS in order to solve multiple complex goals submitted to the knowledge base. There have been numerous studies performed to integrate knowledge bases and databases; see (6 - 10) for a few. Leveraging this technology would allow the global users to define their own knowledge base system for their own needs. Systems for a comptroller, SPO chief, and vice president may be constructed the same but would have different inference engines and would access, for the most part, different facts within the GDBMS. See Figure 11 for a generic description of a proposed knowledge base system.

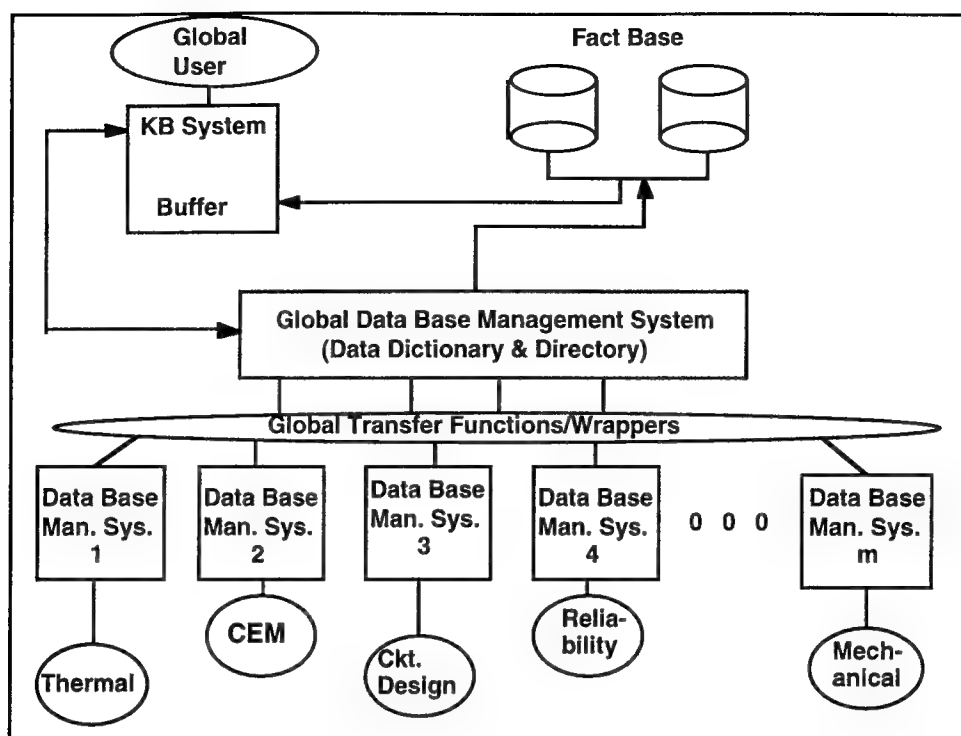


Figure 11. A Integrated Knowledge Base System

The fact base could be obtained from the GDBMS by periodically performing standard queries to update the facts within the knowledge bases. The facts would be stored locally within the knowledge base system (KBS). When a goal is submitted to the KBS the KBS would perform its inferencing on the rules while a second process would determine, based upon the goal, which facts will be needed to meet the goal's needs. Some of the facts will be stored within the primary memory of the machine and the rest of the facts will be buffered into memory as they are required. In this fashion the KBS will perform as if the total fact base is contained within main memory. The fact base can be updated periodically at a rate dependent upon the update dynamics of the heterogeneous databases.

Beyond knowledge there is intelligence or the "capacity to apprehend facts and to reason about them". Computer scientists have been performing research in developing intelligent software and/or agents for years. In particular the Defense Advanced Research Project Agency's (DARPA) David Gunning, is quoted in (11) as saying, "The Defense Advanced Research Project Agency's I\*3 (Intelligent Integration of Information) program is developing advanced technology to provide easy access to information--in the form needed by end users and high-level applications--by

intelligently retrieving, filtering, extracting, integrating, and abstracting information from the growing morass of available data. I\*3 technology is enabling the creation of large-scale, intelligent applications by providing the technology to transform disperse collections of heterogeneous data sources into virtual knowledge bases. These knowledge bases will integrate the semantic contents of those disparate sources to produce integrated information products--in the right form and at the right level of abstraction--for end-user applications. The goals of the I\*3 program are to:

- create new information--integration technology to enable a new level of capability for human and computer users to semantically search, query, monitor, and update large collections of heterogeneous data; and
- develop a suite of information--integration tools to reduce the cost of developing, maintaining, and evolving these large-scale integrated systems."

Intelligent Agents (IA) are Artificial Intelligence (AI) tools that appear to have intelligence by performing functions on their own for a user. They may search databases and extract data, reports, articles, etc. based upon a user's previous defined search criteria. It may go off on its own and perform queries to a database and look for data trends that occur or alert the user of data trends that exceed some threshold. Some of these systems are programmed to perform specific tasks, some learn by observing how a user performs his or her functions within the databases and to attempt to change their behavior to match the user's. These types of intelligent agents could sit "on top" of the KBS and the GDBMS to perform their functions.

In Figure 12 a design is shown of an IA that interfaces to the global user, the KB System and to the GDBMS. It can take direction for example from a global user for automatically searching of the GDBMS to develop reports or search and gather statistics from the raw data and alert the user if parameters exceed pre-defined bounds (e.g., the mean or standard deviation exceeds some value). Different AIs will perform different tasks based upon the global user. For instance, the comptroller's IA will generate different reports and statistics than the IA for the project engineer or the president of the firm. The IA is directly connected to the GDBMS. It requires this connectivity to perform periodic queries to the GDBMS in order to generate its reports and findings for its user. The IA is also directly connected to the KB System. It requires this connectivity for searching the data and knowledge base for creating its reports and findings for the user and also to learn by observing how its user queries and creates "rules" for the KB System. In this manner it can learn what is important and how to better use the KB System rather than just the GDBMS to perform its functions.

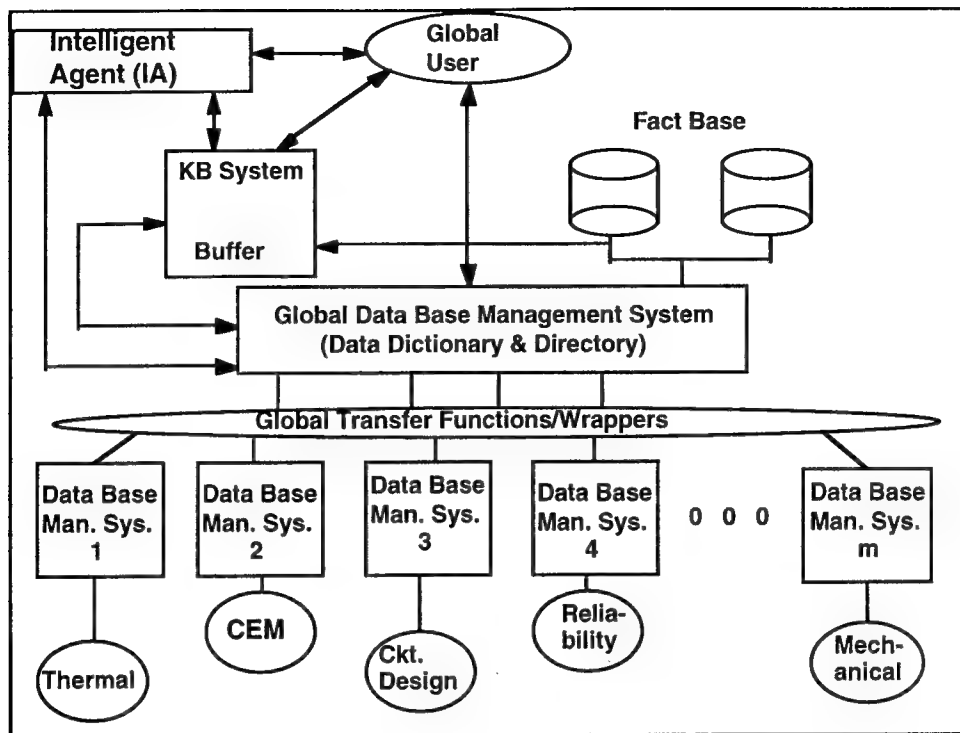


Figure 12. A Data, Knowledge, and Intelligent System

As the KBS and IA evolve, new rules will be added, users will search for new trends, and therefore new facts will be obtained from the GDBMS. Also as new data are added to the domain databases and obtainable through the GDBMS the different KBSs and IAs may update their rules and algorithms, add new rules and algorithms, and change old rules and algorithms. This evolution of rules, algorithms, and data must be coordinated throughout the developers of the domain databases, the DBA of the GDBMS, and the numerous global database, knowledge base, and IA users. Without this control, data consistency and accuracy will be lost. Accurate data is a corporate resource and inaccurate data is a corporate expense. Coordination and configuration control of the corporate database is absolutely necessary. As the users remove themselves from the data by multiple levels of abstraction it is more difficult to maintain information, knowledge, and data lineage.

It was mentioned earlier that the technology for implementing these proposed architectures is available. They can be implemented using a computer network approach with distributed databases located in numerous locations. At each location there can be local area networks with remote client terminals gaining access to their local databases and remote databases via the connection of computers and networks. This approach is beneficial

if portions of what is proposed already exists. If the majority of the databases is not already connected in a distributed fashion, then it is recommended that the intranet and browser approach be seriously considered (12). It provides machine independence via Java and browsers, open database connectivity, and the user interface with which global users are already familiar. In addition, with the coming of the Virtual Reality Modeling Language (VRML) specification, users will be able to send, receive, and interact with renderings of complex drawings from CAD/CAM tools on multiple platforms. It will also allow for collaborative users to interact in real time on the design of complex entities while sharing data, graphics, voice, VRML renderings, and video.

## **7. Conclusions**

The implementation of an integrated architecture of distributed heterogeneous databases as discussed above has many benefits. It allows users to obtain information based upon controlled and accurate data and knowledge. The intelligence obtained is based upon consistent data and knowledge. It should reduce cost through the reduction in the number of databases that will be maintained. It will also increase the number of accurate knowledge bases with an inherent low maintenance cost because of its distribution and coherency. It will also provide more timely, consistent, and accurate data, knowledge, and intelligence which will be accessible by different global users. In addition it provides information lineage, in that there is a direct linkage to intelligence, knowledge, and data. One will know or can derive where and how a result was obtained.

The implementation of this architecture or one that can perform a similar functional capability is recommended for implementation. The benefits are many whether the architecture is implemented as a distributed client/server paradigm or an intranet paradigm using browsers or both. The major conclusion is that it should be built and built based upon currently used CAD/CAM tools, simulation, and analysis models/tools. It should be a "bottom up" database driven system. The technology is here. The benefits are many. The time is now.

## References

- \*1. 1. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Phase II Basic Final Report," CDRL A003, Draft A, March 1995.
  - \*2. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Reference & Specification Document," CDRL A005, Draft A, March 1995.
  - \*3. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Design Tool Integration Document," CDRL A006, Draft A, March 1995.
  4. <http://www.dbms.mfi.com/pccase.html> contains a 1996 DBMS Buyer's Guide.
  5. King, D. and O'Leary, D., "Intelligent Executive Information Agents," *IEEE Expert Intelligent Systems & Their Applications*, Vol.11, No. 6, December 1996, pp. 30-35.
  - \*6. Lazzara, L.V., Marcinkowski, J. M., Capraro, G.T., and White, R.C., "Knowledge Based and Database System Integration: A Base Line for a Design Methodology," RADC-TR-86-132, September 1986.
  7. Lazzara, L.V., Marcinkowski, J., Capraro, G.T., Luizzi, R., and White, R., "A Framework for Extending Information Systems With Knowledge Based Processing Capabilities," *Proceedings of the Computer Software and Applications Conference*, October 1986.
  - \*8. Lazzara, A.V., Marcinkowski, T.M., Tepfenhart, W., Capraro, G.T., White, R.C. and White, T.M. "Data Architecture Concepts for Knowledge Based Systems, KBS and DBMS Integration," RADC-TR-88-245, October 1988.
  - \*9. Lazzara, A.V., Marcinkowski, J.M., Decker, J.M., White, R.C., White, T.M., Capraro, G.T., Darras, D., and Cheung, C., "KBS-DBMS Integration Analysis and Design Alternatives," RADC-TR-87-91.
  10. Capraro, G.T., Cheung, C., and Liuzzi, R. A., "Large Knowledge Based Systems: An Efficient Approach," *Proceedings of the Annual AI Systems in Government Conference*, March 1989.
  11. Hendler, J. A., "Intelligent Agents: Where AI Meets Information Technology," *IEEE Expert Intelligent Systems & Their Applications*, Vol.11, No. 6, December 1996, pp. 20-23.
- \*Items 1,2, & 3 are Distribution authorized to DOD Components only.  
Items 6,8, & 9 are Distrisuiou X - authorized to U.S. Gov't Agencies & Private Individuals or Enterprises eligible to obtain export controlled technical data.



12. Simon, A., "Beyond the Warehouse," *Database Programming & Design*, Vol. 9, No. 13, December 1996, pp. 43-46.

## Appendix A

### Some DBMS Tools

The contents of this appendix were obtained from DBMS 1996 Buyer's Guide at <http://www.dbms.mfi.com/pccase.html>. Not all of the tools that they evaluated are contained here. A selected subset was chosen based upon their evaluations. This is a "living" document that should be revisited before any tools are chosen. The purpose of this appendix is to make the Government aware of the number and types of tools that are commercially available. An assessment of the tools presented here has not been performed by CTI.

#### DB-Examiner

DBE Software Inc

McLean, VA 703-847-9500, 800-760-6940

<http://www.dbesoftware.com>

DB-Examiner analyzes database structures and identifies inconsistencies that adversely affect database integrity and efficiency. It uses advanced algorithms to provide comprehensive analysis and diagnostics. DB-Examiner's relational theory analysis will provide detailed information on all normalization rule violations, referential integrity law violations, and circular relationships that will degrade the quality, efficiency, and flexibility of a database. Its documentation features will produce a full set of reports on tables, constraints, and relationships. DB-Examiner is a Windows client tool that supports Oracle, IBM DB2, IBM SQL/DS, and CA Datacom/DB.

#### Designer/2000

Oracle Corp.

Redwood Shores, CA 415-506-7000, 800-633-0583

<http://www.oracle.com>

A Windows application design solution that incorporates support for business process reengineering (BPR), system analysis, software design and code generation. Through its active repository and integration with Developer/2000, Designer/2000 allows organizations to design and rapidly deliver scalable, client/server systems that can adapt to changing business needs. Developers can develop and deploy applications with Developer/2000, or they can integrate the development process with Designer/2000 to model more complex business solutions. Developers have access to an integrated solution for application modeling and can automatically generate applications by leveraging a common repository. Permits access to a development repository, allowing developers to leverage the BPR models. Also, with the aid of multimedia objects, it gives management greater access to and understanding of evolving business modeling practices. With Designer/2000, an executive can click on a box and listen to audio of the service phone call, or full motion video of a customer representative meeting with a client.

EasyCASE Professional 4.2 for Windows: Workgroup Edition

Evergreen Software Tools Inc.

Redmond, WA 206-881-5149, 800-929-5194

<http://www.estl.com>

A full-featured Computer Aided Software Engineering (CASE) tool that provides complete support for structured analysis and design using a wide selection of structured methodologies for process, data and state-event modeling. EasyCASE supports methodologies by Yourdon-DeMarco, Gane & Sarson, Ward-Mellor (real-time), Yourdon-Constantine, SSADM, Chen, Martin, IDEF1X, etc. using dataflow diagrams (including real-time), state transition diagrams, structure charts, entity-relationship diagrams, and more.

Features: chart editor, multi-user, data dictionary, reports, and model analysis. Includes: data dictionary maintenance utility.

GroundWorks

Cayenne Software Inc. (A Bachman and Cadre Company.)

Burlington, MA 617-273-9003

<http://www.cayennesoft.com>

Windows-based business-modeling software designed to support modeling projects. With data modeling, process modeling, and object-oriented constructs such as entity methods and attribute derivations, it can help build the foundation users need to go forward with client/server plans.

InfoModeler 1.5

Asymetrix Corp.

Bellevue, WA 206-637-1504, 800-448-6543

<http://www.asymetrix.com>

Enables the database professional to create a database schema using English facts and examples. Represents an implementation of object-role modeling (ORM), a methodology popularized at the University of Queensland by Dr. Terry Halpin. ORM provides the ability to assign a wide variety of rules and constraints for business rules, triggers, and stored procedures. Automatically maps the conceptual model to an optimally normalized relational schema creating entities, attributes, relationships, indexes, business rules, triggers, stored procedures, and check clauses. Generates a database definition language for specific targeted databases including: Oracle7, Sybase System 10, Microsoft SQL Server, Microsoft Visual Basic, Access, Microsoft Visual FoxPro, Borland Paradox for Windows, and Borland dBASE for Windows.

Logic Works AOS (Application Object Server)

Logic Works Inc.

Princeton, NJ 609-514-1177, 800-783-7946

<http://www.logicworks.com>

A workgroup model management system, which, working with AOS-enhanced versions of the ERwin database design tool, promotes the sharing and structured management of models. AOS makes data models available directly from a central server, or ModelStore, so all

members of the development team work with the most current models. Differences between archived versions can be reviewed and the current model can be selectively rolled back to previous versions at any time. Keeps a log of all changes made within models, allowing for impact analysis of changes made by users. Using Intelligent Conflict Resolution, multiple users can change a model concurrently. Any conflicts will be identified so that a decision can be made as to which changes will be made to the model. Independent model merge allows ERwin models to be consolidated into one model. Model access and update control is managed through an integrated, flexible security system.

#### Open Workgroup Repository (OWR)

Manager Software Products Inc.

Lexington, MA 617-863-5800, 800-737-6748

A client/server-based set of repository tools that comprise the workgroup tier in MSP's three-tier repository architecture. Provides complete meta data management services on top of facilities provided by its supporting RDBMS. Runs with Oracle, Sybase, Informix, Teradata, and DB2/2. OWR tools operate in Unix, Windows, Windows NT, and OS/2 environments, supporting communications protocols including TCP/IP, Novell, Banyan Vines, and LAN Manager. The repository engine supports ANSI/FIPS standards for repository architecture with a number of useful extensions. The tool suite consists of graphical modeling and management tools for repository administrators and end users. A modeler builds and prototypes the Repository Information Model in a cache file separate from the database. This permits analysis and further prototyping before populating the live repository. Also available is CASE Integrator, a utility that supports the bridging, exchange, and migration of upper CASE design metadata among different CASE tools and the OWR.

#### Silverrun Professional Series

Computer Systems Advisers Inc.

Woodcliff Lake, NJ 201-391-6500, 800-537-4262

<http://www.silverrun.com>

A multiplatform data analysis and design tool comprising four modules, which can be integrated or used separately. The modules operate under Windows, OS/2, Macintosh, and Solaris with interfaces to relational databases including Informix, Progress, DB2, Oracle, and Sybase, and code generation through third parties such as NewEra, Open Environment Corp., Delphi, Object Studio, Synon 2/E, Progress, Omnis7, Uniface, SQLWindows, and PowerBuilder. The four modules include Entity Relationship Expert (ERX), Relational Data Modeler (RDM), Business Process Modeler (BPM), and Workgroup Repository Manager (WRM). ERX offers an embedded expert system that helps modelers create correct, normalized, data models from data structures, existing file definitions, and business rules, and can be used as a reengineering tool. RDM includes automated functions that help

ensure the production of accurate, high-quality database designs, and generates schemas for 16 RDBMSs and application development tools. Object-oriented extensions are available to define actions or tables and specifications on columns. Provides sub-schema support. BPM is a process design tool for ensuring high quality diagramming and documentation, and accurate production of process flows. WRM coordinates the Silverrun toolset and supports the consolidation and sharing of dictionary information in project repositories during system development.

Smarter 2.0

Knowledge Based Systems Inc.

College Station, TX 409-260-5274

<http://www.kbsi.com>

An information, data modeling, and SQL generation tool for Windows that automatically generates SQL code for database implementation, and imports SQL for reverse-engineering of databases into representative data models. The product's ODBC interface provides the flexibility to forward- and reverse-engineer databases from all major database tool vendors. The Validate Model option verifies that completed diagrams are valid information or data models. Smarter checks for connectivity, primary-key usage, duplicate names, unnamed links, and nonspecific relations. Its windows facilitate rapid development, editing, and analysis of model information: the View window is a graphical representation of a standard ER diagram; the Entity/Entity Matrix window is a spreadsheet-like interface showing the interaction between entities via relation links; the Entity/Attribute Matrix window illustrates the use of attributes within entities, including keys and migrated attributes; and the Model Nodelist window offers an expandable outline format for displaying all the entities and views in a single model. Information is collected and stored in reusable pools for efficient creation of multiple models within a single project.

## Appendix B

### ICE Database System Requirements

The ICE design as shown below consists of more than one DBMS integrated into two sets of clients each operating against two levels of servers that communicate and share data in an integrated fashion. One level server executes the Global Data Base Management System. It acts as a server to the global users and integrates the databases from the functional and support model database management systems. The second level server consists of the host computers that execute the support and functional model database management systems. The client systems are executing the database management system(s) that the global users and the functional and support model engineers are using to execute their codes. The following set of requirements was developed to provide a high degree of capability for addressing the numerous functional models, support models, and deployment simulation and emulation tools and models.

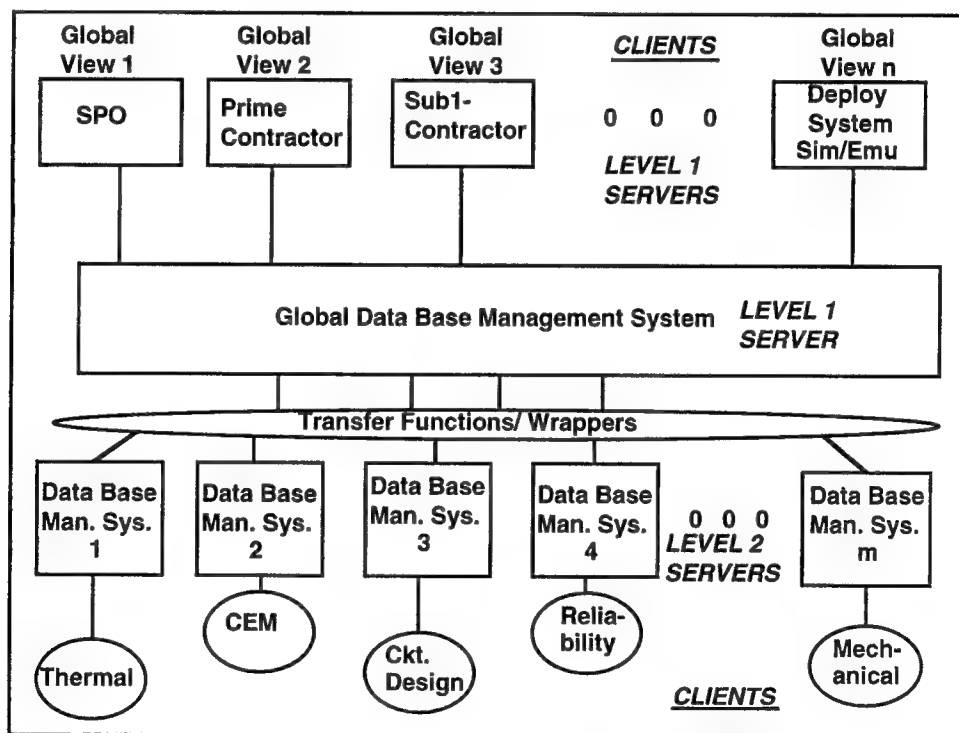


Figure B-1. Integrated Heterogeneous Databases  
Some of the key requirements for the ICE DBMSs are:

- Machine Independence: Client DBMSs must be capable of running on Mac, PC, and Workstation architectures.

- Data and Software Compatibility: Queries, forms, reports, and application software generated on one DBMS must be compatible with different DBMSs operating on different platforms.
- Security: The DBMSs must be capable of operating within classified networks up to the secret level.
- Data Protection: The DBMSs must provide for data protection, e.g., password protection to portions of the databases, data update protection during a power outage, and database backup capability.
- Replication: The DBMSs must support replication of data down to the relation or table level.
- Concurrency Control: The DBMSs must provide control for handling both local (on the same server) and distributed (over a network of servers) data. For example, it must prevent a user from changing data which another user is changing.
- Data Integrity: The DBMSs must provide a data integrity checking capability; e.g., a Date field cannot have a thirteenth month and a Salary field cannot be recorded as an imaginary number.
- Data Accessibility: Data must be accessible simultaneously over networks by multiple users.
- Configuration Control: The DBMSs and related software must provide flexible and programmable configuration control of files.
- User Interface: The DBMSs must provide standard graphical user interface (GUI) utilities to easily build, tailor, and maintain user interfaces (e.g., Input Form Construction), for computer-novice personnel. These interfaces must allow for data entry, data retrieval, DBMS maintenance, and the execution of application software.
- Report Generation: The DBMSs must provide a report generation capability.
- Data Formats: DBMSs must be able to store and retrieve all types of formatted data, e.g., text, numerical, drawings 2&3-D, color, "code" as data, movies, pictures, sound, spreadsheet data, IGES file type data, and multimedia data.
- Utilities: Software utilities must be provided to assist in database development, maintenance, configuration control, and

merging with other databases, e.g., data definition, importing databases, and exporting databases.

- Compliance: The DBMSs must be SQL compliant
- COTS: Commercial Off The Shelf (COTS) software must be used as much as possible. Note that runtime versions of some COTS DBMSs are available for unlimited distribution at no additional cost per seat.
- Licensing: The licensing of the DBMS related software must be such that the purchaser has free access to the source code if the developer relinquishes its maintenance.
- Performance: The system must perform well without undue waiting by a user at all levels of processing.
- Scalability: The DBMSs must be scalable to run on multiprocessor computers.



## **Appendix C**

**October 1995**

### **INTEGRATED COMPUTATIONAL ENVIRONMENT (ICE) FIRST INTERIM REPORT**

#### **Abstract**

This is the first interim report documenting the results obtained in performing Contract F30602-95-C-0109. A brief description of the Integrated Computational Environment (ICE) is provided along with discussions of related Rome Laboratory efforts attempting to solve portions of the ICE challenge. An assessment of database technology for integrating heterogeneous databases and data warehousing is presented along with conclusions and recommendations for building ICE.

## 1. Introduction

The Integrated Computational Environment (ICE) is a new approach for designing and modeling components, boards, boxes, line replaceable units (LRU), subsystems and systems for the USAF. The Department of Defense (DoD) is slowly moving towards the use of modeling and simulation techniques for fulfilling part of the functions that have been performed by military specifications and testing. The old approach was based upon the premise that if each component met the military's specifications then when the full system was integrated it would meet the military performance and environmental conditions. This approach in many cases led to over-designed components and increased costs because the commercial market did not require these designs and could not afford the extra quality. This new trend of using commercial parts, when shown feasible through analysis, modeling and simulation, should bring the cost of military systems down by making use of less costly commercial off the shelf (COTS) hardware and software.

To implement this new approach within the DoD is in itself a challenge. The challenge lies on many fronts, from procurement policies, to testing, to maintenance, to military rights of ownership of data. This particular contractual effort is concerned with the challenge of integrating the different modeling and simulation tools such that Concurrent Engineering (CE) can be performed using these tools and thereby reducing the cost of procuring military systems.

This is the first report within this contractual effort and will cover the first task and a portion of the second task. The first task is to:

... "Review the state of practice of how the USAF and their contractors presently build systems and their up-upgrades. The purpose is to gain enough data and understanding such that any ICE realization will not adversely affect the present approach. The results of this task shall be delivered in the first Interim Report in accordance with the contract schedule."

The second task is to:

... "Research and review programs within the DOD that may be addressing subsets of ICE (For example the Advanced Research Project Agency (ARPA) has a program called Rapid prototyping Application Specific Signal Processors (RASSP).) and programs that are trying to integrate database management systems, heterogeneous software applications, and heterogeneous graphical user interface codes (e.g. Microwave/Millimeter-wave Advanced Computational Environment - MMACE). ICE should be designed to take advantage of what the Government has or will develop in the near future. The

results of this research and review shall be delivered in the second Interim Report in accordance with the contract schedule."

The above tasks were slightly modified because of the changes that occurred from the time the statement of work was completed and the onset of this effort. The first task changed to evaluating the different USAF approaches that have been evolving at Rome Laboratory and the second task has for the most part stayed intact.

The rest of this report presents our findings to date. The following section provides an overview of the ICE and the motivation for its existence. The third section provides a description of those projects within Rome Laboratory that are directly related to ICE. This is followed by an in-depth discussion of the Research Engineering Framework (REF) portion of the MMACE program. The REF is the software initiative that allows for the sharing and integration of varied computer application tools, their data, and IGES files generated by different CAD tools. The fifth section and the Appendix contain a short tutorial on Database Management Systems and the integration of heterogeneous databases. This material is provided to help the reader understand some of the database terminology and the state of the technology for integrating heterogeneous databases. The last section presents our conclusions and recommendations for work that still needs to be performed.

## **2. Overview**

The Rome Laboratory is developing technology to help design and build new or improved weapon systems with the highest reliability, compatibility, and maintainability while using commercial components and minimizing costs. The military procurement process for purchasing systems with military specifications and standards will be changed over the next few years. Methods to integrate commercial components into military systems will rely heavily on computer modeling and simulation as opposed to standards and testing.

There are, however, several sources of inefficiencies and inaccuracies in the current way of using modeling and simulation in the acquisition of DoD systems. The DoD simulation and modeling tools that are available have been developed by many different technologists/disciplines, with each model and its data related to their particular areas of expertise. In addition, the people concerned about reliability, compatibility, and maintainability normally are not involved early in the design process. When they are involved, they are sometimes evaluating data and designs that have been changed. Also, there are data incompatibilities in data attributes, data formats, data values, etc.

An ultimate goal in solving these problems and inefficiencies is to define a unified design and implementation of an Integrated Computational Environment (ICE). This computational ability must provide a consistent and obtainable database, describing an overall system, its components, and its environment, and must provide the capability of integrating Government and commercial data, modeling, and simulation tools. The ICE solution should be transparent to the current tools and methods that are in practice. However, it should provide the compatible framework for integrating the different databases, tools, models, and simulation packages, such that well defined interfaces can be established and controlled so that a more efficient, timely, and accurate exchange of data can occur. A conceptual vision of ICE is shown in Figure C1.

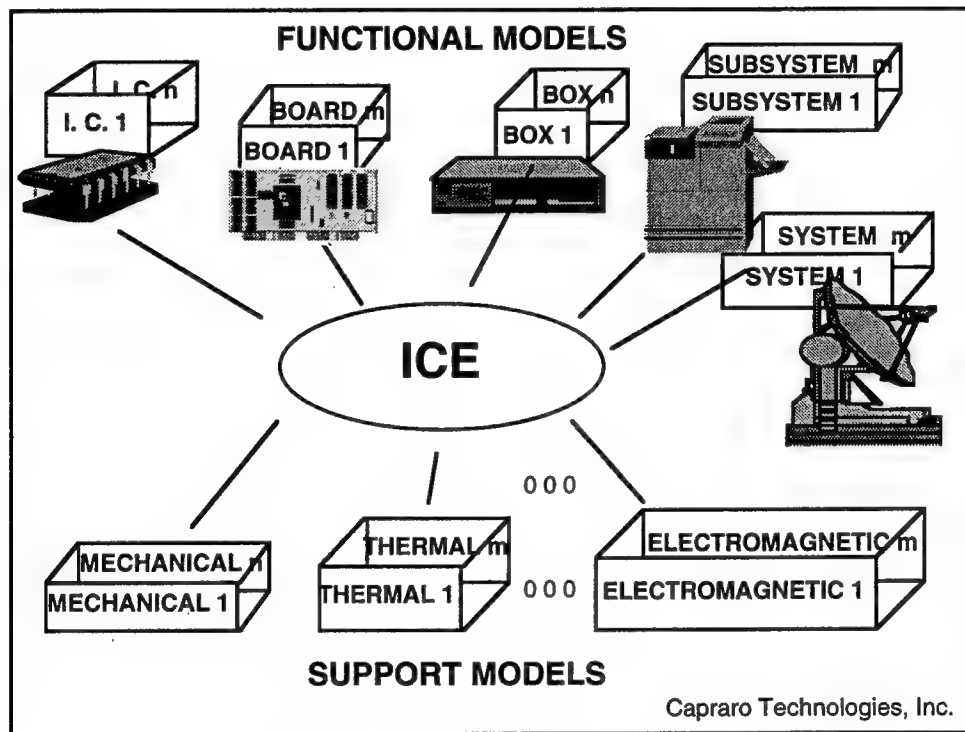


Figure C1. Conceptual View of ICE

A description of this integration of models and tools that can be used over the life cycle of a system is very detailed and will change as the technologies change. For instance Figure C2 represents the type of data required for different electromagnetic compatibility modeling (EMC) tools throughout the life cycle of a weapon system. As the system matures, the amount of data representing the weapon system increases, the modeling tools are more sophisticated, and the tools' respective results are more accurate. Similar matrices can be developed for other disciplines

like mechanical and thermal. In addition, the performance models and their data requirements are also changing depending upon circuit, board, box, subsystem, or system and their maturity.

	INTEG. CKT.	BOARD	BOX	LRU	SUB- SYSTEM	SYSTEM
CONCEPT FORMULATION	NULL	NULL	NULL	NULL	Antenna & Wire Char.	Freq. Allo- cation
REQUIREMENTS DEFINITION	NULL	NULL	G/B/S*	G/B/S	Antenna & Wire Char.	Same
SPECIFICATIONS DEVELOPMENT	NULL	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation
PRELIMINARY DESIGN REVIEW (PDR)	Ground- ing, Line Widths, SPICE/Ck tCodes	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation
CRITICAL DESIGN REVIEW (CDR)	SPICE/Ck tCodes	Wire Codes, Field Codes, & SPICE/Ck t Codes	IEMCAP, Field Codes, & SPICE/Ck t Codes	IEMCAP, Field Codes, & SPICE/Ckt Codes	IEMCAP & Field Codes	IEMCAP, Field Codes, & Freq. Assign- ment
INSTALLATION AND TESTING	NULL	Wire Codes, Field Codes, & SPICE/Ck t Codes	Same + Testing, Lab., Field, & Injectio n	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injectio n	Same + Testing, Lab., Field, & Injectio n
MAINTENANCE	NULL	G/B/S/R* Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing

\*G/B/S/R Represents Grounding/Bonding/Shielding/Radiation

Figure C2. EMC Life Cycle Data Requirements

An ideal Database Management System (DBMS) structure for integrating both the functional modeling and simulation community and the support modeling and simulation community is shown in Figure C3. (A more detailed discussion of this figure is presented in the Appendix.) The center rectangle represents a database management system that integrates the total data representing a weapon system. The bottom databases and users represent the functional and support models and their users. The management of the databases with strict configuration management

will allow the system to be developed in a concurrent mode, supporting Concurrent Engineering (CE). The top portion of the figure provides different users multiple views of the database, describing the system. Some users may only be interested in viewing the system from a structural perspective, others may wish to access the data based upon reliability and maintainability, some may only be interested in testing or meeting environmental specifications, etc. The benefit of a global database having a homogeneous and consistent representation of the weapon system is tantamount to proper management with minimum expense.

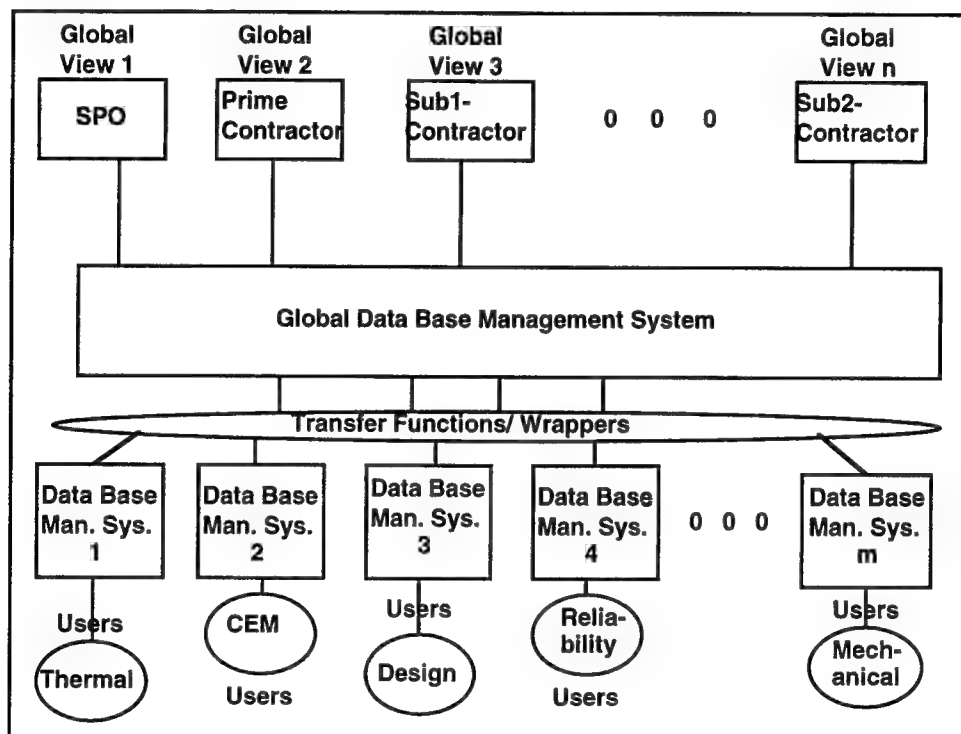


Figure C3. Integrated Heterogeneous Databases

### 3. Rome Laboratory Related Efforts

Within the Electromagnetics and Reliability Directorate of Rome Laboratory there are different thrusts going on that are related to integrating different modeling and simulation tools to achieve a capability similar to ICE. An effort related to a Reliability and Maintainability Information System (REMIS) has pursued some similar challenges that ICE may incur, when trying to integrate different tools' data into a common DBMS. REMIS is to become the standard Air Force database for collecting and processing equipment maintenance data. The point of contact at Rome Laboratory on this effort is Mr. Edward Depalma. A draft document

related to REMIS was reviewed along with a discussion with Mr. Depalma. The Air Force is presently in its second design of the system. The first design didn't work because it was too cryptic in its data values, it was too tough to navigate the database, and it did not make use of a commercial DBMS. The current design is using Structural Query Language (SQL) and the ORACLE DBMS. Mr. Depalma will keep us informed as he receives information regarding their status.

In the mechanical, structural, thermal, vibration, and load analysis areas there are two kinds of modeling approaches being used and pursued at Rome Laboratory. One is based upon Finite Element Analysis (FEA) used by both Industry and the Government. There are 3 widely used FEA codes, NASTRAN, ANSYS and NISA. They are all similar in their data requirements and their output data. The other approach is composed of a set of tools being developed by Rome Laboratory, which is based upon a closed form solution. The tools are new and require less input data from the user. (Point of contact at Rome Laboratory is Mr. Peter Rocci.)

There are 2 approaches in the use of FEA models for entering data that are being employed today. One approach uses standard drafting tools for describing the elements of interest and then the drafting tools' output files are read by the FEA tools. The other approach is to enhance the graphics tools within the FEA codes to describe the elements, thereby not relying on the drafting tools.

As an illustration, let us consider two ways in which an analyst can use a leading drafting and configuration tool called Pro Engineer. This is a drafting tool that allows the designers to share files of all the elements within a system. When a change is made to one element the one file gets updated and all those elements that are affected by the change are notified so that those designers can compensate for the change in their element's design. When their changes are made the process of changes begins again.

The output of Pro Engineer can be used in 2 different ways with the FEA code NISA. First, the NISA analyst can read in the file and use the Model Builder "Display" to subdivide the element, create smaller elements, nodes, and write a file for the computational portion of NISA. This approach requires the user to heavily interact with the code. Second, Pro Engineer generates an output file with the design, establishes the elements and nodes, and generates the material properties, then it writes a file that NISA can then analyze. This approach doesn't use Display and requires less user modeling.

Mr. Bocchi of Rome Laboratory was very helpful in providing the above FEA modeling information and loaning CTI a copy of the NISA

user manual. NISA handles 3-D Geometric input data and allows for Translation, Rotation, and Mirror images. It uses a free field input data stream where fields have a maximum number of characters per input and are separated by commas. It also allows graphical data input. But, more importantly for this effort, it will accept the Initial Graphics Exchange Specification (IGES) formatted data as input. This is important because the MMACE program, referred to in task 2, is based on IGES input files.

A very interesting area of work headed by Mr. Dale Richards compliments and enforces the ICE concept. Mr. Richards released a first version of the Intelligent MultiChip Module Analyzer (IMCMA) computer program in June 1995. It utilizes a blackboard paradigm software system that helps people intelligently define the elements for a FEA thermal analysis of MultiChip Modules (MCM). It was developed by 2 professors at the University of Massachusetts, i.e. Gross and Corkhill, along with in-house Rome Laboratory personnel.

Mr. Richards and his team are also developing a Transmitter/Receiver (T/R) Module Analysis Design Environment (TRADE). This is a superset of IMCMA plus a Finite Element Solution technique developed by Dr. Gross called FEECAP. This tool will handle thermal, electromagnetic, electric, and system reliability.

It appears that Mr. Richards and his team also wish to develop an integrated set of tools feeding off of one database, similar to ICE. The TRADE design is very compatible with ICE. Both MMACE and ICE are a top down design of a global database, using a bottom up direction. The individual tools, representing the bottom (see Figure C3) determine the data requirements for the data managed by the Global Database Management System. The TRADE approach is starting from the physics of the entity under evaluation and attempting to define the database (bottom up) from which all other tools can obtain data. From a system's top down view, TRADE is one of many tool sets requiring data within a structure of a very complicated and large weapon system. A database of this magnitude is a "living entity" and will change and evolve over time as technologies and their models change.

Mr. Richards provided three documents for review: "An Architecture For Intelligent Multichip Module Reliability Analyses", RL-TR-94-71, April 1994 by Daniel D. Corkill of Univ. of Mass., Amherst; "Users Guide To IMCMA" a Rome Laboratory document dated 1 June 1995; and a Draft Final Report entitled "Integrated Finite-Element Generation in the IMCMA Intelligent Multichip Module Reliability Analysis System", by Daniel D. Corkill of Blackboard Technology Group, Inc. Amherst, Mass. The first document provided a background of the prototype IMCMA model and its components, made up of: the Blackboard Technology Group Inc.'s Generic Blackboard



(GBB) software written in LISP; the Sandia National Laboratory's FORTRAN tools for wire meshing written in FORTRAN; integrated FORTRAN tools which perform the finite-element generation; and the nine Knowledge Sources written in LISP which help the user define and model the MCM chip development and well placement.

The second document provides the input and output details for executing IMCMA Version 1. The document is well written and informative. However, the authors claim that they had to perform many software "gyrations" because they were not allowed to change the Sandia code. The last document describes IMCMA Version 2. This version removed the Sandia code. They implemented the software themselves with "tighter" software and knowledge sources. They also added a 3-D capability for visualizing the meshed chips.

The integration of the numerous tools within ICE is based upon integrating the different tools through their input and output data requirements. The management of these data will be performed by a DBMS. To construct a database for a DBMS, a data model must be developed before a database is implemented. To meet this requirement for computational electromagnetic (CEM) Mr. Siarkiewicz has an on going effort to have a SQL definition and schema design performed for the General Electromagnetic Model for Complex Systems (GEMACS) computer model. This database model will probably be first to test the ability of the MMACE's Research and Engineering Framework (REF) to accept a modeling area outside of the Microwave Tube Industry models.

#### **4. Microwave and Millimeter-Wave Advanced Computational Environment (MMACE)**

The MMACE program exists as a Tri-Service and NASA initiative to improve the power tube design process. This program will provide the microwave and millimeter-wave tube industry with an integrated design, simulation, prototype, and manufacturing software environment. It is composed of two portions. One portion is composed of the tube-specific codes and tools that are used to perform the design and analysis of power tubes. The second portion is the Research and Engineering Framework (REF) which contains the programming interfaces, standards, and tools to aid in the integration of the codes and tools. A diagram of the REF is shown in Figure C4 and the reader is directed to the following for more detail information (1,2,3). The REF was studied in detail through these documents and numerous meetings with Mr. Siarkiewicz and a visit to Raytheon Corporation, the prime contractor for MMACE.

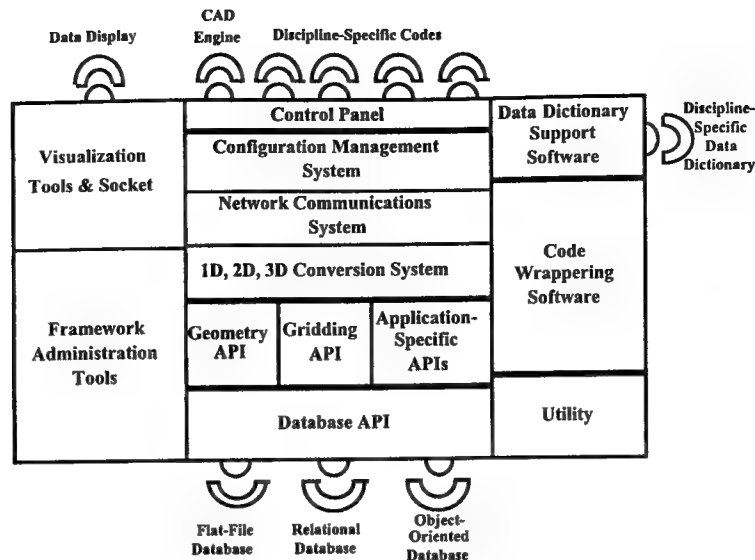


Figure C4. REF Elements for MMACE

The purpose of this in-depth study was to determine if the REF was suitable for handling ICE's requirements. This section provides a description of the current REF design as it applies to ICE's needs. It should be stated that since the REF is still under development, this assessment will be on-going throughout this effort.

The REF is being developed for a well defined set of requirements, for a small industry, and with a limited budget. Many of the developer's decisions were based upon a tight set of constraints. This did not allow the developers to base their design on commercial off the shelf (COTS) tools e.g. a DBMS or Graphical User Interface (GUI) software. Many of the tools and models are built in FORTRAN and the software is developed for a workstation computer with a UNIX operating system. The following description will provide our current understanding of the REF.

Figure C5 shows an overview of the REF's ability to interface with a user and the population sequence of the integrated database. The tube industry has a finite set of analysis tools which apply to the different stages or portions of a microwave/millimeter-wave tube. Each of these tools require IGES input files, both geometry and parametric data, for them to operate. The user can describe the portion of the tube using a Computer Aided Design (CAD) package that generates IGES files. Because different CAD packages generate different IGES compatible files for the same design, the REF developer chose a subset of the NASA standard IGES file description to store in the database. This required them to develop an IGES translator that can read the output of a commercial CAD package and convert it to a standard such that data incompatibilities would not exist between different CAD packages

different CAD packages operating on the same design entity. They have written these software tools to operate with the AutoCad and Pro Engineer CAD programs. They claim that adding other CAD programs is not difficult. The output of the IGES translator constitutes the major portion of the REF database.

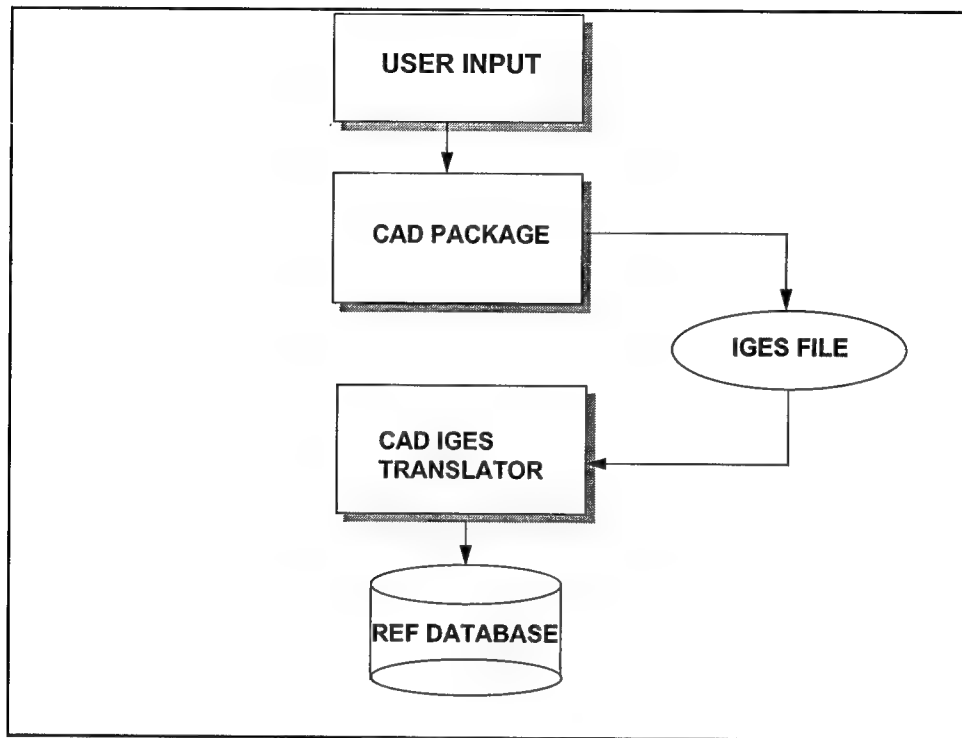


Figure C5. REF Database Population Sequence

Figure C6 depicts how the database interacts with the different MMACE modeling tools or codes. The database partitions the geometry data and the parametric or properties data into two classes of files. These files however, are not linked at the database level at this time. They can be linked however, by the user changing the data at the CAD package level. Each modeling tool or code can interface with the REF database through a code wrapper that exercises the Geometry Application Programmer Interface (API) software developed by the Raytheon team. Geometry data can only be read from the database at this level. The wrapper, however, can read and write parametric data. The output from the codes will also be stored within the database and be accessed by the different codes. The writing of this software has not been completed at this time.

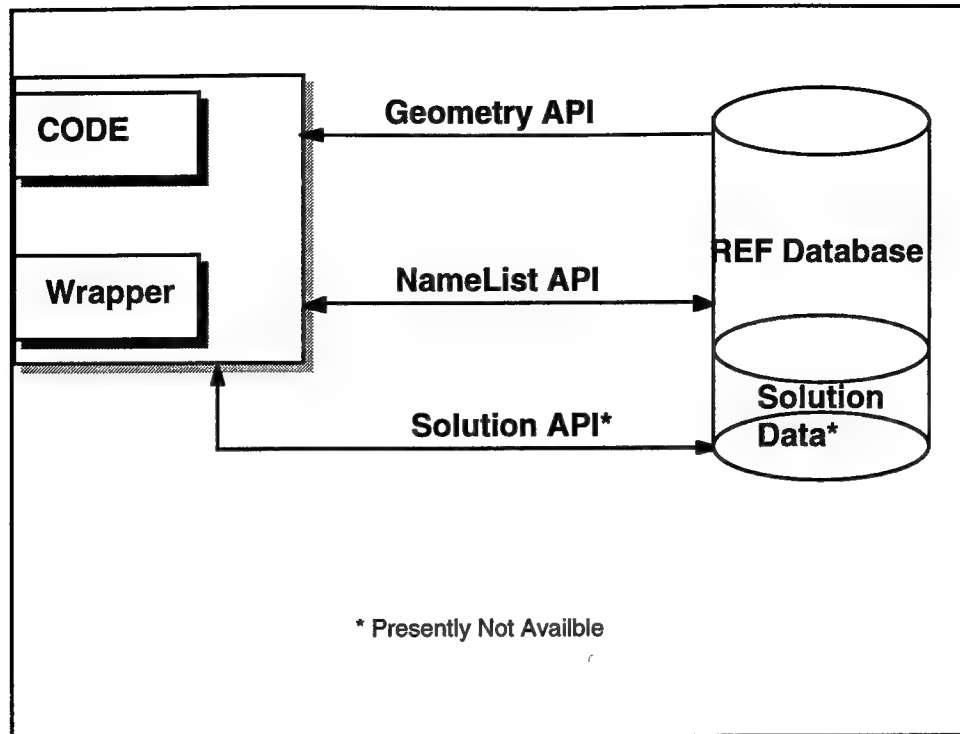


Figure C6. REF Database and Code Interface

The integrating of the different data required by the different codes is performed mainly through two approaches. The geometry data parameters are controlled through the use of the CAD packages and their IGES file formats. The naming conventions and/or parametric data are controlled by the tube industry through committee. That is, each code has access to and must stay compliant with a fixed set of parameters, units, names, etc. This forces the community to have a homogeneous database with few parameters that are code dependent, i.e. lie outside their common intersection. Figure C7 depicts a subset of the codes, where each set in the Venn Diagram represents a tube code and its input parameters. Few attributes (or input parameters) are code dependent and not shared. The four codes identified are those that have unique attributes to describe the model. The Shared Data (center set) is accessed by eight or more codes and does not have any code dependent attributes.

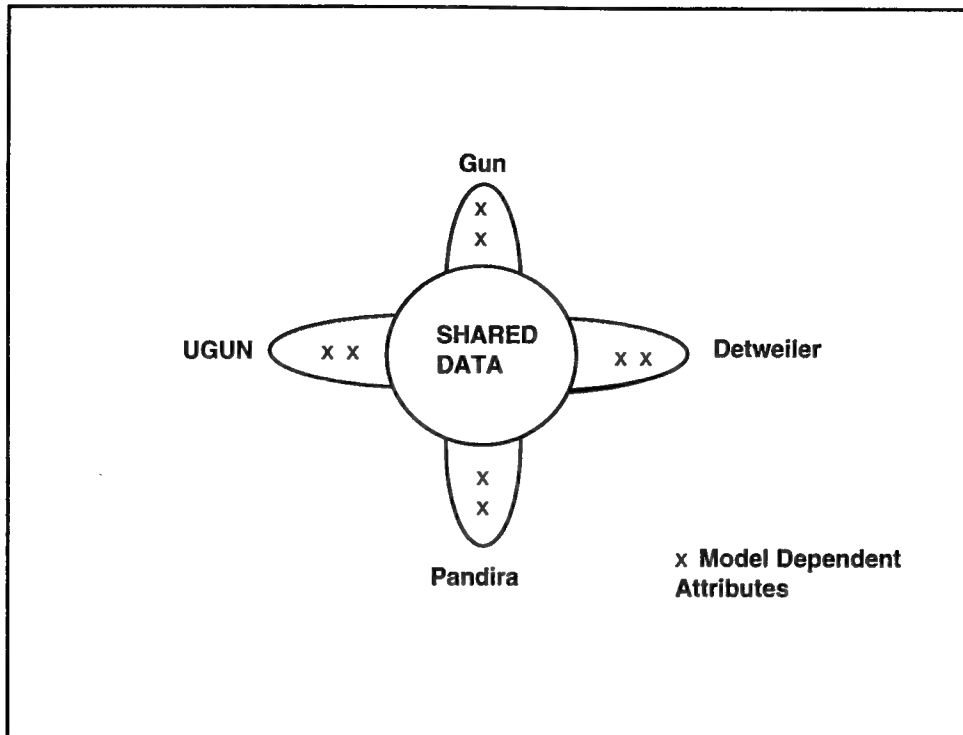


Figure C7. REF Database and Code Relationships

The REF also has a Data Dictionary (DD) capability which maintains a list of the attributes within the database. A DD within a DBMS, stores meta data and authorization information, such as key constraints and user privileges and is the direct interface to the database. (Meta data are those data about the data, e.g. an attribute's name, field type, and size of field.) The REF DD, however, does not interface with the REF database. Changes to the DD do not affect the database and changes to the database are not reflected in the DD. The DD within the REF only performs a bookkeeping function that allows one to query which attributes are in the database, but it is not capable of searching the database for the values of these attributes. The DD is as up to date as the industry manually maintains its contents. This is important since, adding new data to the database is easy. However, changes to the database affect the DD and all wrappers interfacing models to the database. Therefore, the industry must manually update the wrappers and the DD when one adds, deletes, or changes the database schema or design. This manual process could be simplified if the DD and the database were implemented with a DBMS. This would provide data independence from the application tools and the wrappers and would minimize the cost for maintaining the system.

## **5. Integrating Databases**

The ICE design will include the integration of heterogeneous databases. There is work going on in the DBMS community related to integrating heterogeneous databases and the warehousing of databases. For the reader to understand Figure C3 and appreciate this important area a brief overview of the technology and current research efforts is provided in Appendix A. The material covers an overview of DBMS technology, data models, standardization, major components of a DBMS, and present research approaches for integrating heterogeneous databases and the warehousing of databases. These different approaches that the industry is considering will be helpful in designing an ICE architecture.

## **6. Conclusions and Recommendations**

A purpose of this effort is to design ICE to take advantage of what the Government has or will develop in the near future. We are not done with the second task, but our interim results are encouraging in attempting to meet this design requirement. At this time we envision two different approaches that require further investigation. These are to pursue the integration of heterogeneous databases and the REF as the glue for integrating the multiple applications and tools using an architecture similar to Figure C3 as shown in Figure C8. This approach will integrate DBMS technology at the Modified REF level and at an Enhanced REF level. The Enhanced REF and a DBMS will perform the Global glue as described above. A second approach is to have the TRADE as the integration architecture as shown in Figure C9. The key to these approaches is to take advantage of what has been accomplished in the integration of databases and what has been developed by the Government.

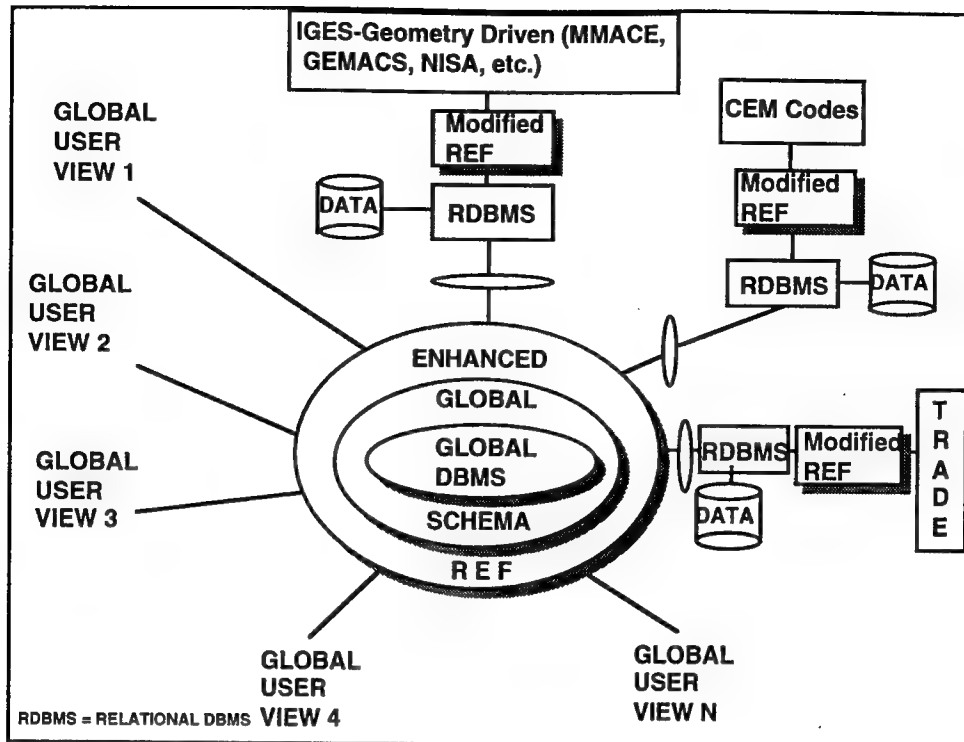


Figure C8. A REF/DBMS-based ICE Approach

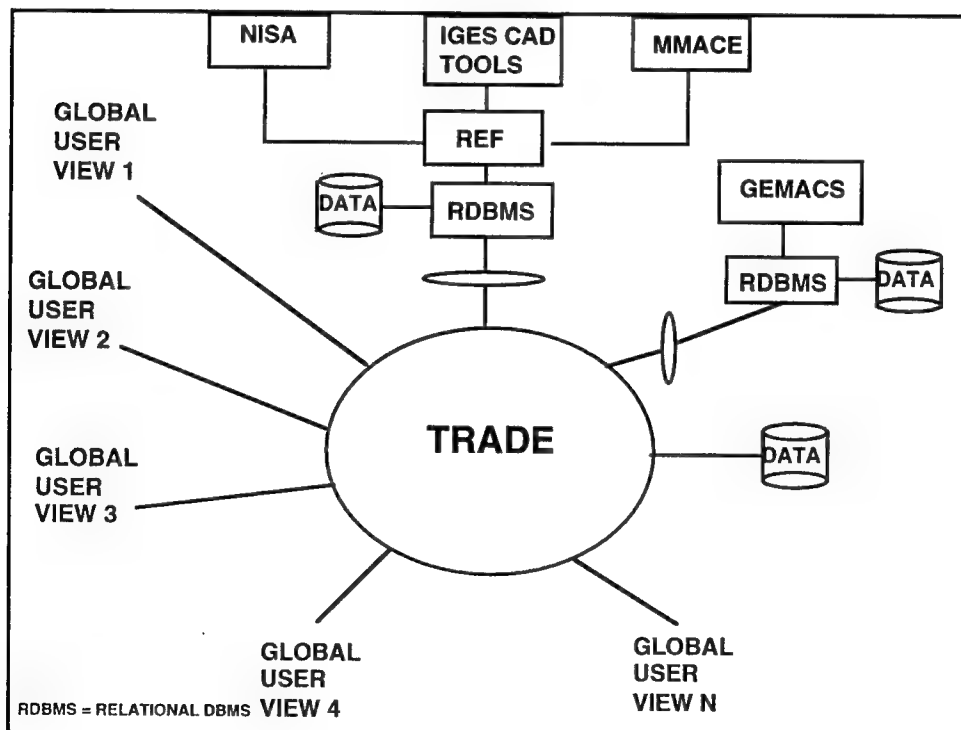


Figure C9. A TRADE-based ICE Approach

The first approach makes use of the general technology afforded to the DBMS area and the tool integration advances made by the REF. It provides for Data Independence, Data Maintenance, Data Security, Conflict Resolution, Data Portability, Data Dictionary and Directory, etc. of the DBMS area. It also adds the REF's capability of integrating parametric and IGES file based tools built with different software development environments. One concern with this approach is to find a DBMS that can manage the global configuration management requirements of CE.

The second approach, having TRADE as the glue for integration, is considered because it can accept both GUI generated data and parametric data. It has a built-in object oriented data handling ability within its blackboard framework. The extent and capability of its processing ability as compared to a commercial DBMS is unknown. Its CE ability and its ability for integrating and accepting disparate tools operating with different DBMSs is also unknown.

In evaluating each of the approaches it should be noted that the tools will have to interface to a DBMS of choice and these DBMSs will have to interface with the Global entity, whether its TRADE, REF, or a commercial DBMS, yet to be identified. In addition, the integration of analysis tools should be easy, and the access and storage of data should be as seamless as possible.

It is recommended that an Enhanced REF and TRADE be investigated thoroughly for determining their potential role as the global integrator. Commercial DBMSs should be evaluated and identified for their applicability. Commercial tools for integrating numerous DBMSs should be investigated. The Modified REF design requires further study as the interface architecture for accepting IGES generated files from any tool and to interface these data to a DBMS. A Modified REF should include maximizing the usage of DBMS functions included in commercial systems.



## Appendix C1

### Integrating Databases

To understand the technology that is available and to recognize the challenge of integrating different DBMSs, it is necessary to understand some of the basic terminology of DBMSs and the different schemes or schemas that have evolved over the years. (See references 4, 5, and 6.) A Database Management System (DBMS) consists of a collection of interrelated data and a set of programs to access that data. A major purpose of a DBMS is to provide each user with an abstract view of the data. In Figure C10, an example of three levels of views seen by different users is depicted. The Conceptual View is that view of the total database as modeled by its developer and Database Administrator. The User Views seen at the top of the figure represent those subsets of the database as seen by different users. Only a few select users normally see the total database as does the Database Administrator. The Physical View is the representation of how the actual data are partitioned and maintained both within memory and on secondary and tertiary storage.

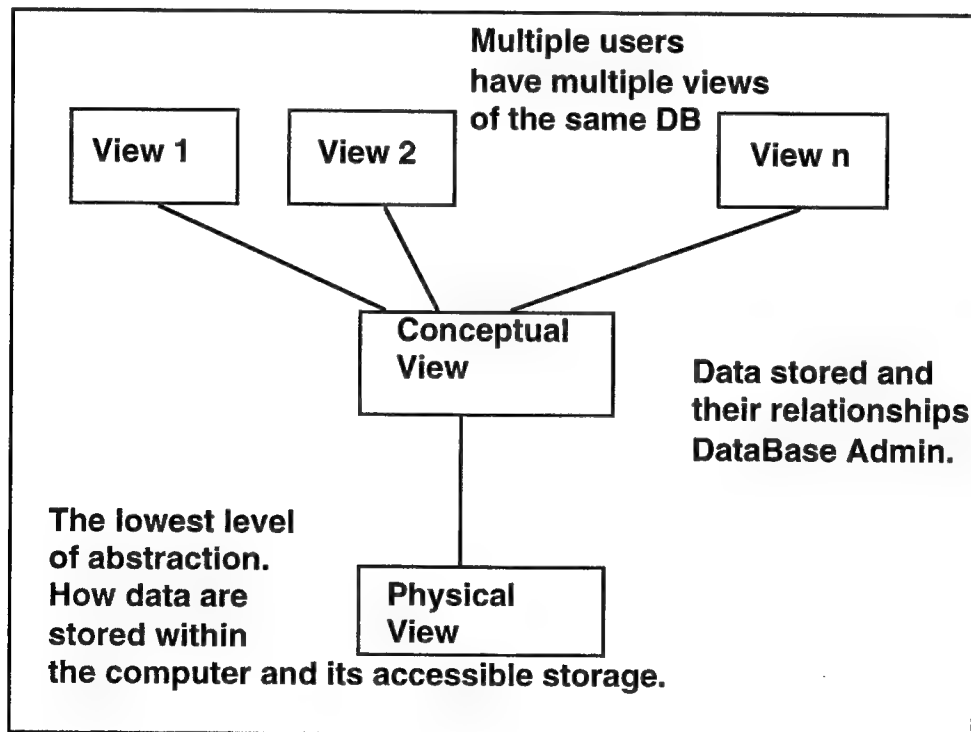


Figure C10. Three Level View Model

The following models that are briefly described are used by both the users and the Data Administrator for depicting portions or all of the databases. DBMSs hide certain details from the users in

order to simplify their interaction with the system. Since some of these users are not programmers, they have no need to see how data are both logically and physically manipulated within the computer and its accessible storage. However, these details are crucial when trying to integrate two databases whether they are built using the same DBMS, the same data model, or a different model. Different ICE-integrated analysis code databases may represent the same entity different ways. In order to integrate them we must understand, for example that mean time between failure in one database is in years and in another it is in months. It is for these database reasons that each database must be described at the SQL level as presently being performed for GEMACS, in order to determine how to keep ICE attributes consistent. It is also important to recognize how the data are viewed and stored, so that software can be designed and built to retrieve the same attribute stored differently in two or more DBMSs having different data models.

The following data models and system structure are described to communicate an understanding of the detailed data that will be required in order to integrate databases that have different data models, different DBMSs, and are resident on different computers.

### **Data Models and the Object-Based Logical Model**

A Data Model is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. There are 3 Categories of Data Models: Object-Based (O-B) Logical Models, Record-Based Logical Models, and Physical Data Models. Two of the widely used O-B Logical Models are: Entity-Relationship (E-R) Model and the Object-Oriented Model. The E-R Model is based on the perception that the real world is made up of entities and the relationships among them. An entity is an object that is distinguishable from other objects by a specific set of attributes. A relationship is an association among several entities. The sets of all entities of the same type and relationships of the same type are termed an entity set and relationship set, respectively. In addition there is a mapping cardinality constraint which expresses the number of entities to which another entity can be associated via a relationship set. (The following examples were obtained from (6)).

A relationship between each entity Customer and his or her entity Account is CustAcct. The collection of all the relationships is a Relationship Set. See Figure C11 for an example.

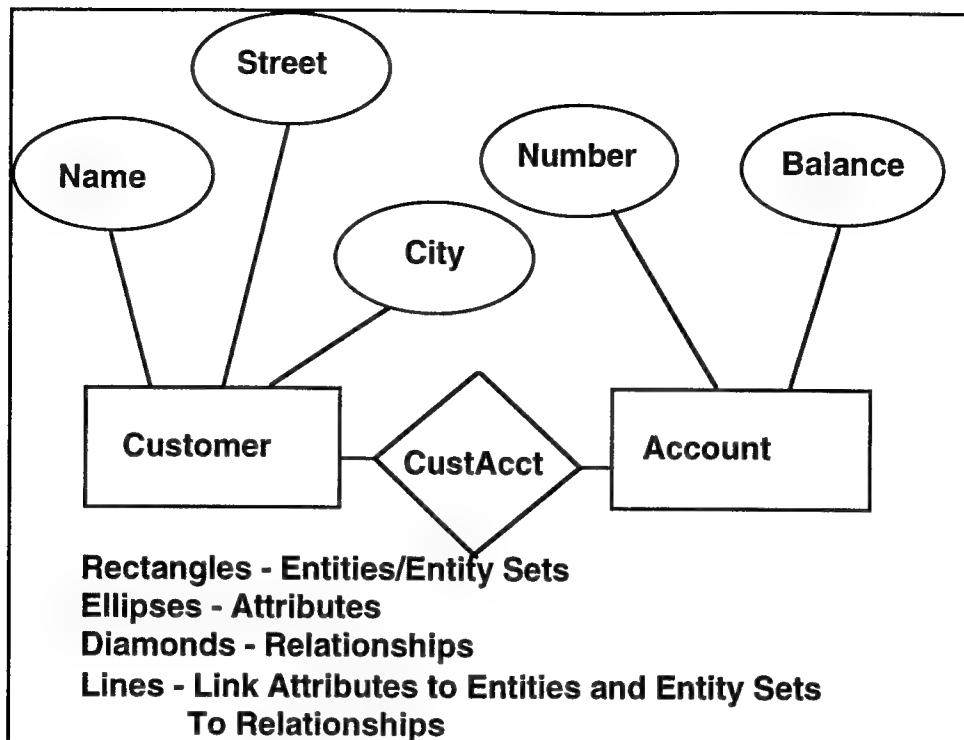


Figure C11. An E-R Diagram

The Object-Oriented Model is based upon a collection of objects. An object contains Instance Variables, whose values are themselves Objects. An Object can contain Objects, which can contain Objects, which can contain Objects, ... An Object can also contain logic or code as an instance variable, which are called Methods. Objects that contain the same types of values and the same methods are grouped together into Classes. Accessing an Object is via a Message or sending a Message which invokes a stored Method. The call interface of the Methods of an object defines its externally visible part. The internal Instance Variables and Methods are not visible externally. This results in two levels of data abstraction. See Figure C12 for an example of a Bank Account Object.

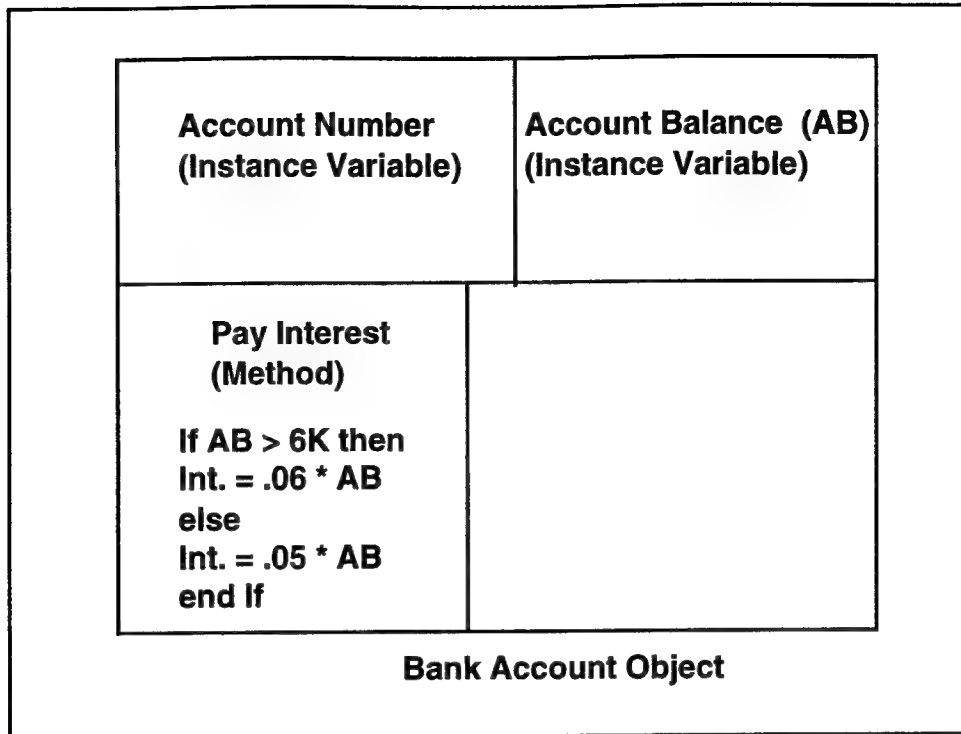


Figure C12. An Object Diagram

Changing the interest computation, in this example, only involves changing the Method. Note: unlike in the E-R model, each object has its own unique identity independent of the values it contains, i.e. two objects containing the same values are distinct. The distinction among individual objects is maintained in the physical level through the assignment of distinct object identifiers.

### **Record-Based Logical Models.**

Like Object-Based Models, Record-Based Models are used for describing the Conceptual and View Levels. They are used to specify overall logical structure of the database and to provide a higher-level description of the implementation. The database is structured in fixed-format records of fixed number of fields (or attributes) and each field is usually of fixed length. Note that fixed length is convenient in modeling and managing the Physical-Level implementation of the database. The Object-based model which allows for an arbitrary depth of nesting of objects results in variable-length records at the physical level. Note also that Record-Based models do not allow for the integration of data and code. The three most widely accepted models are the Relational, Network, and Hierarchical.

The Relational Model represents data and relationships among data by a collection of tables, each of which has a number of columns with unique attribute names. See Figure C13 for an example where name, street, city, balance, and number are attributes in two different relationships. Each row in a table represents an occurrence of a relationship and are sometimes called records. Two relationships can be joined together based upon their common attributes and their respective values. Note from the example that Shiver and Hodges share an account number (647) that has a balance of \$105,366.00.

<u>name</u>	<u>street</u>	<u>city</u>	<u>number</u>
Lowery	Maple	Queens	900
Shiver	North	Bronx	556
Shiver	North	Bronx	647
Hodges	Sidehill	Brooklyn	801
Hodges	Sidehill	Brooklyn	647

<u>number</u>	<u>balance</u>
900	55
556	100000
647	105366
801	10533

Figure C13. Relational Tables

The Network Model, is viewed as a collection of records and the relationships among them are represented by links (which can be viewed as pointers). The network's structure can represent any complex or simple graph. An example is shown in Figure C14 using the same two records presented in Figure C13. The linkage between the two records within the relational model was created by replicating the same attribute and its values in both relations. In the Network Model the attribute only appears within one relationship once and the DBMS creates and maintains an explicit connection between the two relationships using links.

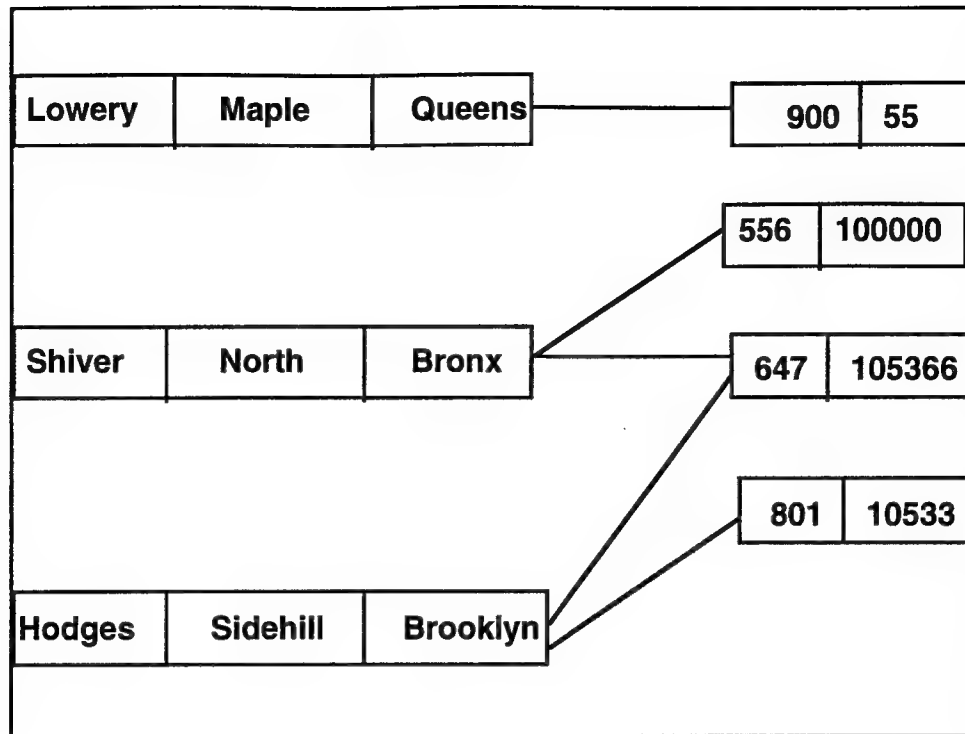


Figure C14. A Network Diagram

The Hierarchical Model is similar to the Network Model. It has Records and Links. It differs in that the records are organized as a collection of trees rather than arbitrary graphs. Figure C15 provides an example.

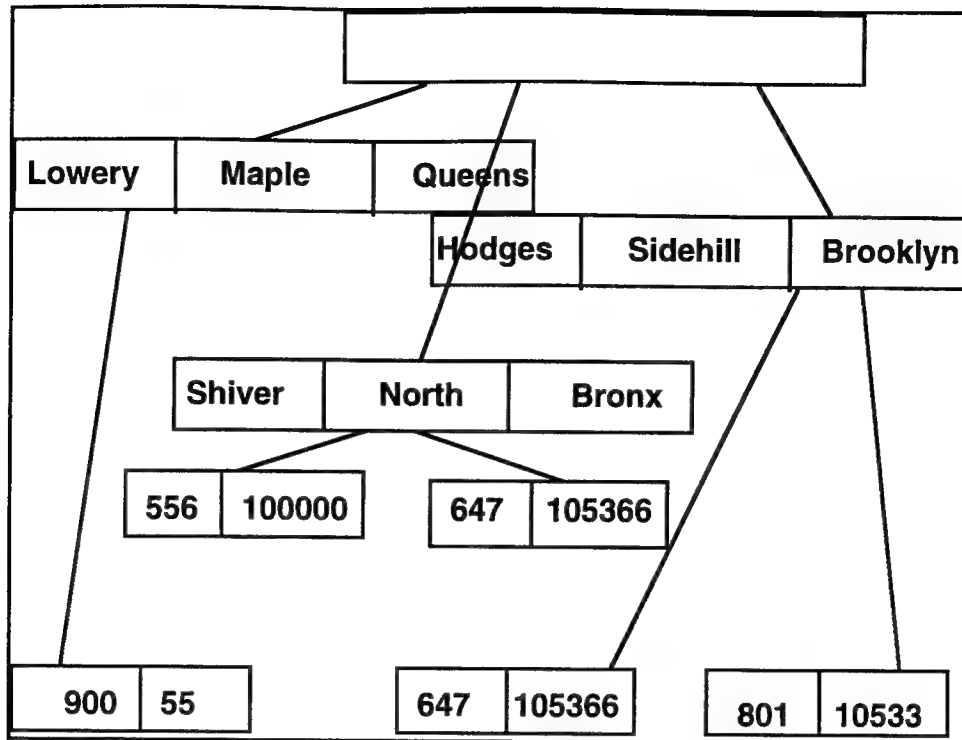


Figure C15. A Hierarchical Diagram

The Relational Model differs from the other two models in that it does not contain explicit pointers but uses implicit pointers to maintain relationships between records. This freedom from the use of pointers and an extension to set theory has allowed for a formal mathematical foundation to be defined for the relational model.

### Physical Data Models

The Physical Data Models describe the data at the lowest level and their description is not necessary for our discussion.

### Overall System Structure

The performance of a DBMS depends on the efficiency of the data structures used to represent the data in the database and on how efficiently the system is able to operate on these data structures.

The basic functions of a DBMS is built upon those functions or basic services provided by the computer's operating system. A DBMS consists of a number of components and data structures that operate "on top of" the operating system. Figure C16 provides a typical System Structure.

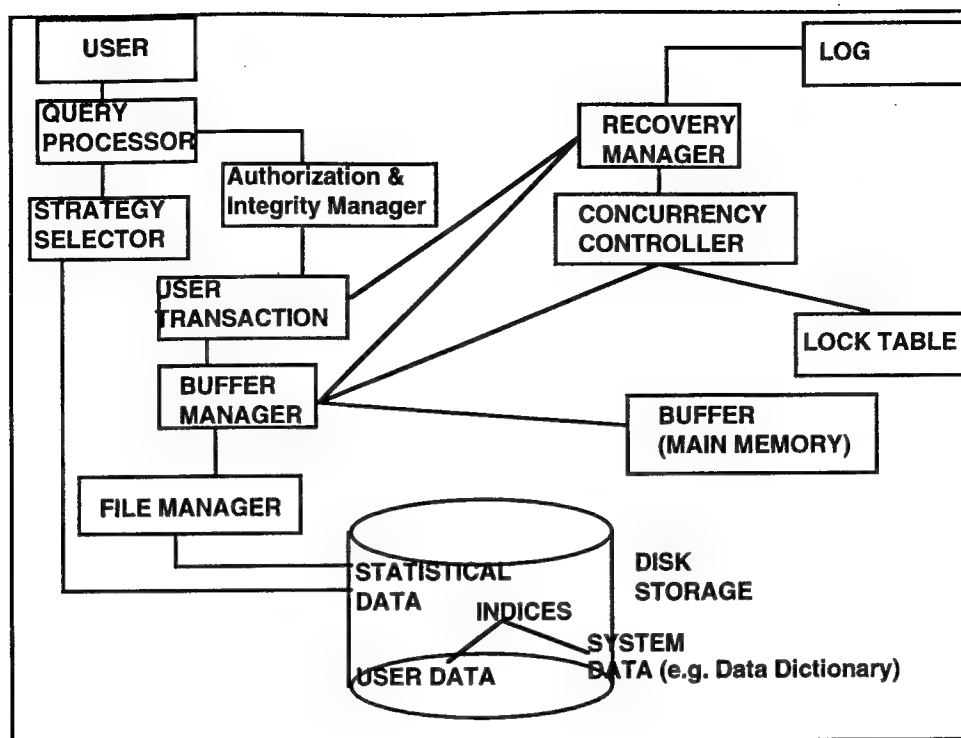


Figure C16. DBMS System Structure

#### Database Components:

**File Manager:** Manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

**Buffer Manager:** Responsible for the transfer of information between disk storage and main memory.

**Query Processor:** Translates statements in a query language into a lower-level language.

**Strategy Selector:** Transforms a user's request into an equivalent but more efficient form for executing the query.

**Authorization and Integrity Manager:** Tests for the satisfaction of integrity constraints and checks the authority of users to access data.

**Recovery Manager:** Ensures the database remains in a consistent and correct state despite system failures.

**User Transaction:** That part of main memory allocated to each user transaction for the storage of copies of data items.



Log: A main memory buffer area that holds records before being written to stable storage.

Concurrency Controller: Ensures that concurrent interactions with the database proceed without conflicting with one another.

Lock Table: A portion of main memory that maintains data as to which transactions have control over which data within the database.

#### Database Structures (Stored within disk and/or main memory.):

Data Files: The database itself i.e. the user's data.

Data Dictionary: Stores meta data and authorization information, such as key constraints and user privileges. Meta data are those data about the data, e.g. attribute's name, field type, and size of field.

Indices: Data elements used for fast access to data items holding particular values.

Statistical Data: Stored data about the data in the database, used by the strategy selector.

#### **Standardization**

Since the advent of DBMSs they have progressed from the hierarchical and network approaches to the relational model. The relational model and its underlying mathematical foundation has led to the development of numerous DBMSs and presently is the primary data model for commercial data processing applications. (This status may be changing in the near future with the surge of Object-Oriented Databases.) However, because of their wide acceptance and sheer numbers there has been a need to have different DBMSs share data, i.e. import and export data between each. This has occurred through standardization.

Originally called Sequel, the Standard Query Language (SQL) was implemented as part of the System R project of the 1970s by IBM. In 1986 ANSI published a SQL standard. It is considered as THE standard relational database language. It has several parts:

Data Definition Language (DDL),  
Interactive Data Manipulation Language (DML),  
Embedded Data Manipulation Language (designed for use with PL/I, COBOL, Pascal, FORTRAN, and C),  
View Definition,  
Authorization,

Integrity, and  
Transaction Control.

The basic structure of SQL consists of three clauses: **select**, **from**, and **where**.

The **select** corresponds to the projection operation (lists the attributes).

The **from** corresponds to the Cartesian product operation (lists the relations to be scanned).

The **where** corresponds to the selection predicate. It consists of a predicate involving attributes of the relations that appear in the **from** clause.

Consider the following SQL example related to the relational address table shown in Figure C11.

```
SQL:
Select    name, street
From address_table
Where     number = 900.
```

This example will provide the name and street address of those people whose number is equal to 900.

The above paragraphs provide an overview of some of the key elements in the DBMS field that have been used to build databases. This information is necessary when attempting to integrate two databases built with different data models on different computers. They help define the complexity of the integration process and software required. The basic concern of this study is how does one interface and process those data that are presently located within diverse DBMSs operating within different computers. In general, as shown in Figure C3, we wish to ask queries of a homogeneous database that is really composed of attribute values from heterogeneous databases. To meet this requirement a review of the literature was performed to determine the state of the technology for interfacing or integrating heterogeneous databases that together compose ICE's data for existing tools. An overview of what is appearing in the research literature will be presented. This will provide information about the current state of the art for performing an integration of different databases that are or will manage ICE components.

### **Present Research Approaches**

Consider the diagram shown in Figure C3. This figure represents a generic architecture for integrating databases. (The interested reader is directed to (7, 8, 9, 10, 11) for a detailed overview and collection of papers pertaining to this area.)

Each of the views at the top of the Figure C3 represents different people or organizations that wish to process data that are contained within the DBMSs at the bottom of the figure. These DBMSs operate independently and are maintained by their own staff for their own users who of course have multiple views of their databases. These DBMSs may be hierarchical, network, relational, flat files, or object-oriented databases. The method that allows for a homogeneous approach is based upon an implementation of a Global DBMS (7, 8, 9, 10, 11). This DBMS acts as the interface to the different DBMSs that contain the actual data. It retrieves requests from the Global users at the top of the figure and spawns off queries to the individual DBMSs as if they were one of its own users shown at the bottom of the figure. There are different methods for achieving this capability. Some approaches use a Global Data Dictionary/Directory (DD/D) (9, 10, 11). This DD/D contains all the information or meta data on which attributes are contained within which DBMS, their format, size, file/relation's name, DBMS, synonyms, homonyms, etc. This DD/D is contained within the Global DBMS. When the Global DBMS receives the query from the user it parses it, checks its syntax and determines if the query can be fulfilled by analyzing the request against the data within the DD/D. If all is correct then requests or sets of subqueries are composed and issued to the individual DBMSs. These queries are then interpreted, reformatted and communicated to each of the DBMSs in their own resident query language. This transformation function is represented by the ellipse-shaped lens within Figure C3. Different approaches perform this function different ways. Some perform this function by using transfer functions (9, 10, 11, 12) and some use wrappers (13).

The transfer function approach is bi-directional, it simply transfers the generic query language of the Global DBMS to the language of the specific DBMS. There would be one transfer function for each of the DBMSs. When the query is fulfilled the transfer function then computes its inverse function, i.e. it converts the DBMS's response to a format that the Global DBMS understands. For instance the Global DBMS may use SQL as its query language and a DBMS may use Query By Example (QBE).

The wrapper approach contains the same capability of the transfer approach. It contains some of the same data and logic contained within the Global DBMS and the DD/D. The wrapper can retrieve a request from the Global DBMS in its language and insulates the individual DBMS from the query. The wrapper contains all the mapping functions from the Global request to the individual DBMS and maintains all of the synonyms, format data, etc., in order to retrieve and process any of the Global DBMSs requests.

The above two approaches for integrating DBMSs have had many variations. One approach lets the users of the individual DBMSs have access to the Global DBMSs in addition to their individual

DBMS (14). Another approach that is appearing in the literature is the Data Warehouse (13) approach in which the Global users are not necessarily wanting to just search data within the organization to answer queries that are time dependent and accurate, but are looking to gather information that would otherwise not be known. For example, they wish to perform statistical analysis of the data that are stored over multiple databases and over multiple years.

It is thought by some that the above two general approaches are very time consuming and labor intensive to implement. In addition, as any of the individual DBMSs change their schema representations, change formats of attributes, etc., then the Global DD/D and Wrappers must be updated accordingly. Another approach is to shift the burden of developing and maintaining the Global DD/D and its associated logic by requiring the individual DBMSs to maintain the interface to the Global schema through multidatabase language systems. In this manner each DBMS represents its data to the Global schema through a nonprocedural SQL based language. This approach causes a sharing of responsibility between the Global Database Administrator and the local Database Administrators. However, it loses a level of data independence in its solution.

When the number of databases gets very large it may be difficult to build a very precise DD/D. Bright, et. al. (14) have developed a Summary Schemas Model (SSM) as an extension to multidatabase or heterogeneous database systems, to provide linguistic support to automatically identify semantically similar entities with different access terms. Their summary schema is a concise, more abstract description of the semantic contents of the individual database schemas that compose the heterogeneous databases. Their model uses specific linguistic relationships between schema terms to build a hierarchical global data structure which describes the information available in the databases in an increasingly abstract form. This model would be helpful in building the Global DD/D discussed above.

Another approach being pursued to help in building the interface of heterogeneous databases is based upon a model independent theory for the exchange of data among heterogeneous information or database systems. This is being pursued using Mediators to facilitate the exchange of semantic values (14, 15), where a Mediator is a software model or module that contains the logic for unraveling imprecise user requests. An implementation of this approach (16) is through an extension of SQL called Context-SQL (C-SQL). The approach, when implemented is not normally seen by the user but is processed in the background. Consider the following illustration of semantic values.

1600(Units = 'lines of code',

```
Comments = 'not included').  
2000(Units = 'lines of code',  
Comments = 'included'(Estimated% = '20')).
```

In the first line the value 1600 has two properties: *Units* and *Comments*. In the second line the value 2000 also has two properties: *Units* and *Comments*. However, in this latter example the *Comments* property is also a semantic value having the property *Estimated%*. One can interpret the above data to represent that the number of lines of executable code in both entities is 1600. This approach is complementary to the above approaches. It would be extremely helpful in building and maintaining all of the approaches mentioned above. It also would allow for the building and maintaining of dynamic databases and knowledge bases that make up the heterogeneous databases.

Multiple approaches for Data Warehousing also exist. Consider the following approach shown in Figure C17. This approach by Windom (13) has monitors which are software tools that are capable of identifying changes in the individual information sources (data and knowledge bases) to determine if they should be propagated to the Integrator function. The Integrator function software accumulates the results of the Monitor functions and updates the Data Warehouse accordingly. The Data Warehouse is maintained and accessed with the aid of a DBMS. This approach differs from the above approaches in that the needed data for the Warehouse are known and the Warehouse is "back filled". The requests for data are obtained via a copy of portions of the databases in the Warehouse and the actual databases are not queried in a dynamic state with bi-directional functions. This approach is good for relatively static information sources and the user's needs are predictable for specific portions of the data. Imagine that in the CE development of a system the Data Warehouse database would contain only the current approval version of the system's design.

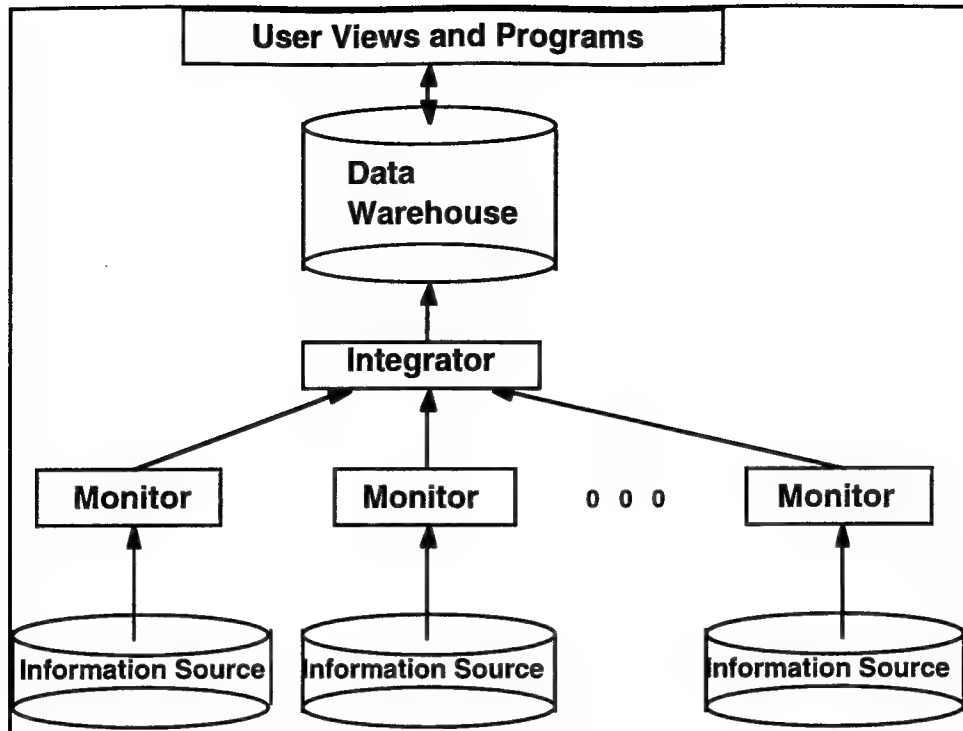


Figure C17. A Data Warehouse Architecture

The approach shown is only one iteration of different ways of obtaining a Warehouse of data. It is different than integrating heterogeneous databases as shown in Figure C3. The difference is primarily predicated on the needs of the Global user. In the heterogeneous DBMS the Global user wishes to form queries or apply transactions against all of the databases in a "real time" mode while viewing the heterogeneous databases as homogeneous. The Data Warehouse is an approach that allows an enterprise to capture, filter, cleanse, and reformat portions of old and current databases such that one can perform decision support, trend analysis, forecasting, statistical analysis, and perform "what if" processing on large amounts of data. The needs of these two approaches are different and yet require similar tools in order to be implemented.

## References

- \*1. Hantman, B., Hicks, R., and LaBelle, J. "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Phase II Basic Final Report", CDRL A003, Draft A, March 1995.
- \*2. Hantman, B., Hicks, R., and LaBelle, J. "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Reference & Specification Document", CDRL A005, Draft A, March 1995.
- \*3. Hantman, B., Hicks, R., and LaBelle, J. "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Design Tool Integration Document", CDRL A006, Draft A, March 1995.
- 4. Date, C. J., *An Introduction to Database Systems*, Addison-Wesley, 1976.
- 5. Martin, J., *Computer Data-Base Organization*, Prentice-Hall, 1975.
- 6. Korth, H. F. and Silberschatz, A., *Database Systems Concepts*, McGraw Hill Book Company, 1986.
- 7 Gupta, A., Editor, *Integration of Information Systems: Bridging Heterogeneous Databases*, IEEE Press, 1989.
- 8. Kambasyasshi, Y., Rusinkiewicz, M., and Sheth, A. Editors, *First International Workshop on Interoperability in Multidatabase Systems*, IEEE Computer Society Press, 1991.
- \*9. Lazzara, A. V., Marcinkowski, J. M., Capraro, G. T., and White, R. C., "Knowledge Based and Database System Integration: A Base Line for a Design Methodology", RADC-TR-96-132, September 1986.
- \*10. Lazzara, A. V., Marcinkowski, J. M., Decker, J. M., White, R. C., White, T. M., Capraro, G. T., Darras, D., and Cheung, C., "KBS-DBMS Integration Analysis and Design Alternatives", RADC-TR-87-91.
- 11. Capraro, G. T., Cheung, C., and Liuzzi, T. A., "Large Knowledge Based Systems: An Efficient Approach", Proceedings of the Annual AI Systems in Government Conference, March 1989.
- 12. Templeton, M., Brill, D., Son, K. D., Lund, E., Ward, P., and Chen, A. L. P., "Mermaid - A Front-End to Distributed Heterogeneous Databases", Proc. IEEE, pp. 695-708, vol. 75, no. 5, May 1987.

\*Items 1,2, & 3 are Distribution authorized to DOD Components only.  
Items 9 & 10 are Distribution X - authorized to U.S. Gov't Agencies & Private Individuals or Enterprises eligible to obtain export controlled technical data.

13. Windom, J., "A Warehousing Approach to Data and Knowledge Integration", A Presentation at the Intelligent Knowledge Base Interchange Workshop sponsored by ARPA and Rome Laboratory, 17/18 Nov. 1994.

14. Bright, M. W., Hurson, A. R., and Pakzad, S., "Automated Resolution of Semantic Heterogeneity in Multidatabases", ACM Transactions on Database Systems, pp. 212 - 253, vol. 19, no. 2, June 1994.

15. Chu, W. W. "(CoBase) Scalable and Extensible Cooperative Information Systems", A Presentation at the Intelligent Knowledge Base Interchange Workshop sponsored by ARPA and Rome Laboratory, 17/18 Nov. 1994.

16. Sciore, E., Siegel, M., and Rosenthal, A., "Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems", ACM Transactions on Database Systems, pp. 254 - 290, vol. 19, no. 2, June 1994.

17. Martin, P. and Powley, W., "Database Integration Using Multidatabase Views", Proceedings of CASCON '93, pp. 779 - 788, vol. 2, Oct. 1993.



## **Appendix D**

**June 1996**

### **INTEGRATED COMPUTATIONAL ENVIRONMENT (ICE) SECOND INTERIM REPORT**

#### **Abstract**

This is the second interim report documenting the results obtained in performing Contract F30602-95-C-0109. An in-depth description of the Integrated Computational Environment (ICE) is provided along with a design of how a portion of the Microwave/Millimeter-wave Advanced Computational Environment (MMACE) program can be used as a foundation for building ICE. A description is provided that demonstrates the integration of heterogeneous databases within the same domain and from multiple domains of interest (i.e. vacuum electronics industry and electromagnetic compatibility).

## 1. Introduction

The Integrated Computational Environment (ICE) is an approach for designing and modeling components, boards, boxes, line replaceable units (LRU), subsystems and systems for the USAF. The Department of Defense (DoD) is slowly moving towards the use of modeling and simulation techniques for fulfilling part of the functions that have been performed by military specifications, and testing. The old approach was based upon the premise that if each component met the military's specifications then when the full system was integrated it would meet the military performance and environmental conditions. This approach in many cases led to over-designed components and increased costs because the commercial market did not require these designs and could not afford the extra quality. This new trend of using commercial parts, when shown feasible through analysis, modeling and simulation, should bring the cost of military systems down by making use of less costly commercial off-the-shelf (COTS) hardware and software.

To implement the ICE approach within the DOD is in itself a challenge. The challenge lies on many fronts, from acquisition policies, to testing, to maintenance, to military rights of ownership of data. This particular contractual effort is concerned with the challenge of designing the integration of the different modeling and simulation tools such that Concurrent Engineering (CE) can be performed using these tools and thereby reducing the cost of procuring military systems.

This is the second report within this contractual effort and will cover the results of the second task. The second task is to:

"Research and review programs within the DOD that may be addressing subsets of ICE (For example, the Defense Advanced Research Project Agency (DARPA) has a program called Rapid prototyping Application Specific Signal Processors (RASSP).) and programs that are trying to integrate database management systems, heterogeneous software applications, and heterogeneous graphical user interface codes (e.g. Microwave/Millimeter-wave Advanced Computational Environment - MMACE). ICE should be designed to take advantage of what the Government has or will develop in the near future. The results of this research and review shall be delivered in the second Interim Report in accordance with the contract schedule."

The above task was slightly modified because of the changes that occurred from the time the statement of work was completed and the onset of this effort. The second task looked at the different related programs and they were reported within the first interim report. This report documents how the MMACE program can be used as a foundation for building ICE.

The first report provided an overview of the ICE and the motivation for its existence. It also provided a description of those projects within Rome Laboratory that are directly related to ICE, a description of the Research Engineering Framework (REF) portion of the MMACE program and a short tutorial on Database Management Systems and the integration of heterogeneous databases.

This second report provides a more in-depth description of the ICE concept. This is followed by a discussion of the REF and how it can be enhanced by hosting some of its elements on a Relational Database Management System. The fourth section contains a description of how the REF structure can be used to integrate heterogeneous databases within a defined domain of interest (e.g. the vacuum electronics industry, the Electromagnetic Compatibility technology area). The fifth section describes how the REF architecture provides the basis for building an integration of heterogeneous databases from multiple domains.

## **2. Overview**

The Rome Laboratory is developing technology to help design and build new or improved weapon systems with the highest reliability, compatibility, and maintainability while using commercial components and minimizing costs. The military acquisition process for purchasing systems with military specifications and standards will be changed over the next few years. Methods to integrate commercial components into military systems will rely heavily on computer modeling and simulation as opposed to standards and testing.

There are, however, several sources of inefficiencies and inaccuracies in the current use of modeling and simulation for the acquisition of DoD systems. The DoD simulation and modeling tools/codes available for system development and deployment were built by many different technologists/disciplines, with each code and its data related to its own area. In addition, the people concerned about reliability, compatibility, and maintainability normally are not involved early in the design process nor in the deployment modeling process. When they are involved, they are sometimes evaluating data and designs that have been changed or they are involved after the system is deployed and is not functioning as designed or expected.

An approach to minimize these problems and inefficiencies is to define a unified design and implementation of an Integrated Computational Environment (ICE). This computational ability must provide a consistent and obtainable database, describing an overall system, its components, and its environment, and must provide the capability of integrating *Government and commercial* data, modeling, and simulation tools. The ICE should be

relatively transparent to the current tools and methods that are in practice. However, it should provide the compatible framework for integrating the different databases, tools, models, and simulation packages, such that well-defined interfaces can be established and controlled for a more efficient, timely, and accurate exchange of data. A conceptual vision of ICE is shown in Figure D1.

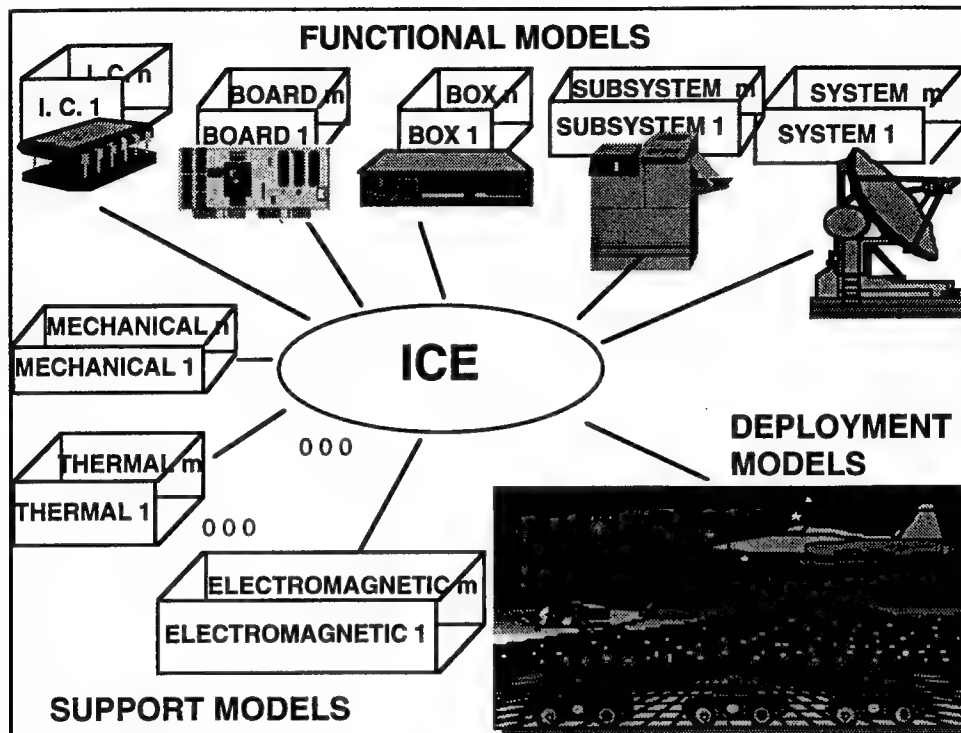


Figure D1. Conceptual View of ICE

ICE supports functional models, support models, and theater-level deployment models. Functional models are those models used to develop the components of a system to meet a system's primary performance requirements. The throughput of a computer, the sensitivity level of a communications receiver, and the radiated power of a radar are examples of system component's primary performance requirements. The support models are those models that are concerned with a component meeting a system's secondary set of requirements. These are usually related to environmental concerns such as mechanical, thermal, and electromagnetic. Theater-level deployment models are related to that process of evaluating new or unavailable components to determine their performance in actual and varied deployment environments. These models may be strictly digital simulations or they may be composed of a mixture of actual components, digital simulation models, and components which emulate other components. With the proliferation of computers within most military systems and the reduced DOD

budget, it is becoming more common for the military to exercise theater-level simulation and/or emulation models to evaluate new or proposed military systems rather than building a prototype system.

The development and deployment process of a new system, e.g., radar, aircraft, or missile, is very complex and involves many people with varied capabilities and objectives. It usually requires a prime contractor and several subcontractors with many people at different locations. These people can be divided into three basic groups based upon their interests. Group 1 consists of those people interested in building a system's components, e.g., high power tubes, processors, amplifiers, sensors, power supplies. An example may be a sub-contractor or a component provider or supplier. Group 2 consists of those people interested by technology or support function, e.g., circuit design people, thermal, electromagnetic, structural, signal processing, communications, radar, contracts, legal, accounting. Group 3 are those people interested in the system-level effects of integrating a system within the deployment environment e.g., system simulations, system emulations, battlefield simulations/emulations. These three groups can be partitioned further by the data required of the computer applications or codes used in an individual's job, e.g., the computational electromagnetic (CEM) area is composed of codes like GEMACS, low frequency codes, high frequency codes, etc.

Consider the potential benefits gained if the data requirements of these different groups were consistent, computerized, secure, and instantly accessible anywhere throughout the world. Connection to a global database from any terminal with a modem would allow for the retrieval of the most detailed data instantly. *This capability would reduce the cost and compress the schedule of system development, deployment, and maintenance throughout a system's cycle, while enhancing performance and safety. The computer technology to accomplish this is here today; but the methods and tools for integrating the data among the three different groups is not in place.* As an example consider Figure D2.

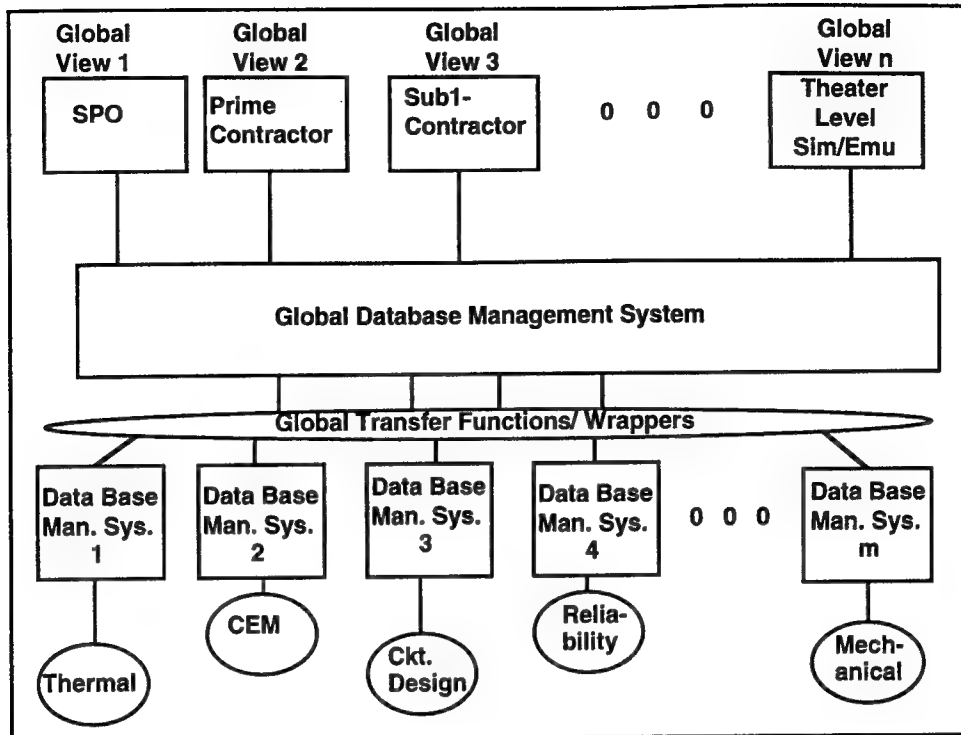


Figure D2. Integrated Heterogeneous Databases

Figure D2 illustrates an approach for integrating a collection of heterogeneous databases from the bottom up. The bottom portion of the above diagram depicts each set of users partitioned by technology (i.e., Group 2). Each user within a technology would have a consistent database that represents any component of interest across all of the codes that are used in that technology over the life of the component. The different databases (thermal, CEM, design, etc.) would be integrated into another consistent database by the Global Database Management System (GDBMS). This allows all users access to the total database whether they are a technology modeler (Group 2), a sub-contractor (Group 1), or a Government agent assessing new technologies in a simulated battle field environment (Group 3). Access to the data within the GDBMS can be obtained within any group given the need to know. The data can be stored at one location centrally located or across a distributed network of computers. Data can be obtained in "real-time" for analysis, meetings, inquiries, and reporting at any location with a computer and a modem.

To obtain a consistent set of data that are available to many throughout the development and deployment of a weapon system, we must begin building a structure based upon existing data that are already being gathered by the respective groups. (See the bottom portion of Figure D2.) In modern-day systems the digitization of data usually takes place when people begin to design the system's

components. They primarily use computer codes accepted by the community and/or company proprietary codes. However, it is the data, not the codes, that drive the requirements for an architecture like that shown in Figure D2. In many organizations the individual users are using their own codes and are not sharing data via a database management system. It is this level of the architecture that must be integrated first. To start the process by defining the data requirements from the users at the top level of the architecture (i.e. the global viewers at the top portion of Figure D2) would be too costly and more importantly would disrupt the current process.

As an illustration of the data involved, consider the EMC community. Figure D3 illustrates the data required by the EMC community for different components and at different stages of a component's development and deployment. The EMC community uses a subset of the codes within the CEM area. The data required by most of the different users within the Groups are dependent upon their codes, the component of interest (e.g., radar, integrated circuit), the acquisition stage, and the deployment environment of the component.

	INTEG. CKT.	BOARD	BOX	LRU	SUB- SYSTEM	SYSTEM	THEATER LEVEL
CONCEPT FORMULATI ON	NULL	NULL	NULL	NULL	Antenna & Wire Char.	Freq. Allo- cation	EM Environ- mental Survey
REQUIREME NTS DEFINITIO N	NULL	NULL	G/B/S*	G/B/S	Antenna & Wire Char.	Same	Power Cull & Propaga- tion Studies
SPECIFICA TIONS DEVELOPME NT	NULL	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation	Antenna to Antenna & Environ- mental Coupling Analysis
PRELIMINA RY DESIGN REVIEW (PDR)	Ground- ing, Line Widths, SPICE /Ckt Codes	G/B/S	Port Emission Levels & Susc. Levels	Port Emission Levels & Susc. Levels	Antenna & Wire Char.	Freq. Allo- cation	Antenna to Antenna & Environ- mental Coupling Analysis
CRITICAL DESIGN REVIEW (CDR)	SPICE /Ckt Codes	Wire Codes, Field Codes, & SPICE/ Ckt Codes	IEMCAP, Field Codes, & SPICE/ Ckt Codes	IEMCAP, Field Codes, & SPICE/ Ckt Codes	IEMCAP & Field Codes	IEMCAP, Field Codes, & Freq. Assign- ment	Inter- platform EM Dynamic Simula- tion/Emu- lation
INSTALLAT ION AND TESTING	NULL	Wire Codes, Field Codes, & SPICE/ Ckt Codes	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing, Lab., Field, & Injection	Same + Testing & Monitor- ing
MAINTENAN CE	NULL	G/B/S/R* Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	G/B/S/R Inspec- tion & Testing	Same + Testing & Monitor- ing

\*G/B/S/R Implies Grounding/Bonding/Shielding/Radiation

Figure D3. EMC Life Cycle Data Requirements

There is one of these matrices for each technology discipline and for each management function (e.g., accounting, contracts, legal). Data integration must begin within each of the technologies. For the most part, the analysts and engineers within each technology presently use different codes and are not integrated nor share their respective data in any computerized efficient form.

The building of the architecture shown in Figure D2 begins by integrating data at the lowest of levels. How does one integrate data required by heterogeneous codes within the same technology



and across multiple engineering disciplines? This area is being addressed in the Tri-Service Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research and Engineering Framework (REF) development program and will be discussed in the next section.

### 3. Research and Engineering Framework (REF)

The MMACE program is a Tri-Service and NASA initiative to improve the power tube design process. It is composed of two portions. One portion is composed of the vacuum electronics codes and tools that are used to perform the design and analysis of power tubes. The second portion is the Research and Engineering Framework (REF) which contains the programming interfaces, standards, and utilities to aid in the integration of the codes and tools. A diagram of the REF is shown in Figure D4, and the reader is directed to references 1-3 for more detailed information.

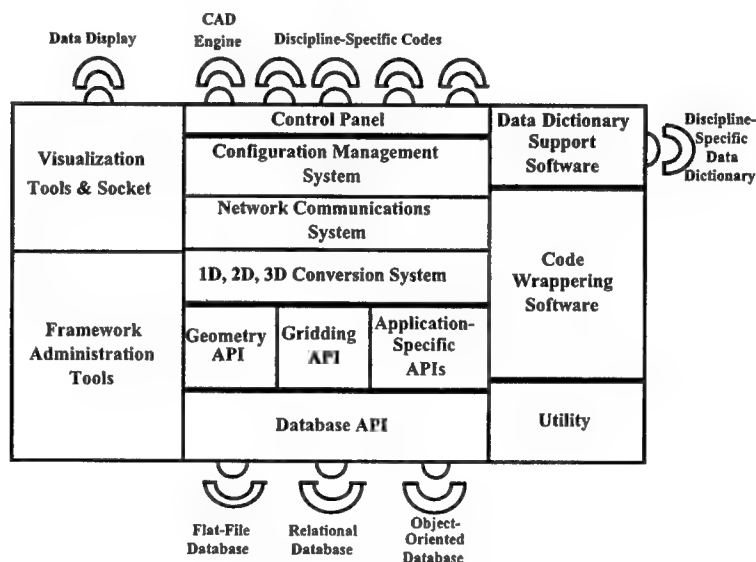


Figure D4. REF Elements for MMACE

The REF is being developed for a well defined set of requirements, for a small industry, and with a limited budget. The following is a brief description of the REF. Figure D5 shows an overview of the REF's ability to interface with a user and the population sequence of the integrated database. The vacuum electronics industry has a finite set of analysis tools which apply to the different stages or elements of a microwave/millimeter-wave tube. Each of these tools requires Initial Graphics Exchange Specification (IGES) input files, both geometry or parametric data, for it to operate. The user can describe the portion of the tube using a Computer Aided Design (CAD) package that generates IGES files. Because different CAD packages generate different IGES-compatible files for the same design, the REF developers

chose a subset of the NASA standard IGES file description to store in the database. This required them to develop an IGES translator that can read the output of a commercial CAD package and convert it to a standard format or specification such that data incompatibilities would not exist between different CAD packages operating on the same computer or entity. They have written these software tools to operate with the AutoCad and ProEngineer CAD programs.

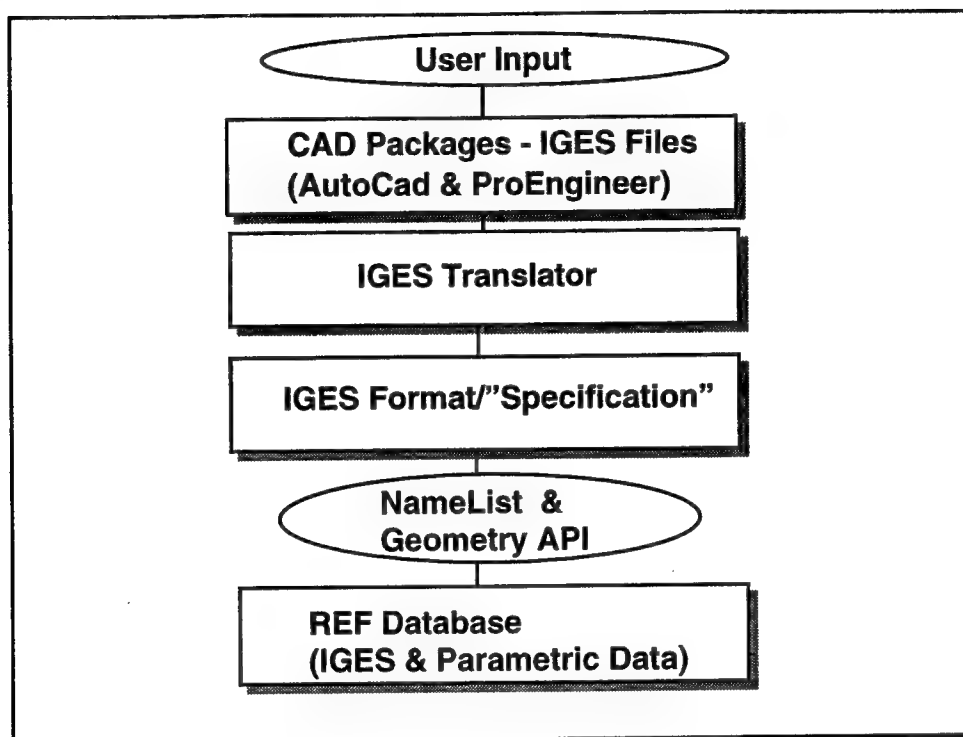


Figure D5. REF Database Population Sequence

The integration of the different data required by the different codes is performed mainly through two approaches. The geometry data parameters are controlled through the use of the CAD packages and their IGES file formats. The naming conventions and/or parametric data are controlled by the tube industry through consensus. That is, each code has access to and must stay compliant with a fixed set of parameters, units, names, etc. This forces the community to have a homogeneous database with few parameters that are code-dependent, i.e., lie outside their common intersection. Figure D6 depicts a subset of the codes, in which each set in the Venn Diagram represents a tube code and its input parameters. Few attributes (or input parameters) are code-dependent and not shared. The four codes identified are those that have unique attributes to describe the model. The Shared Data, the center set or major intersection, is accessed by eight or more codes.

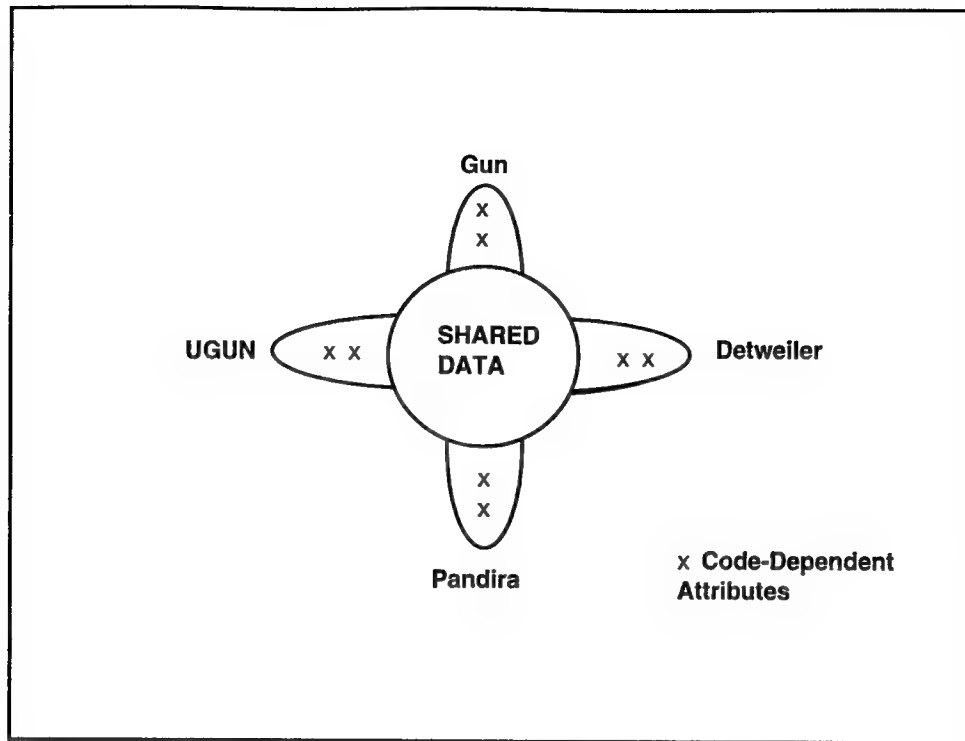


Figure D6. REF Database and Code Relationships

The REF also has a Data Dictionary (DD) which maintains a list of the attributes within the database. A DD within a DBMS stores meta data and authorization information, such as key constraints and user privileges, and is the direct interface to the database. (Meta data are those data about the data, e.g., an attribute's name, field type, and size of the field.) The DD within the REF only performs a bookkeeping function that allows one to query which attributes are in the database, but it is not capable of searching the database for the values of these attributes. The DD is as up-to-date as the industry manually maintains its contents. This is an important issue since adding new data to the database is easy. However, changes to the database affect the DD and all wrappers interfacing codes to the database. *The industry must manually update the wrappers and the DD when one adds, deletes, or changes the database schema or design. This manual process could be simplified if the DD and the database were implemented with a DBMS. This would provide data independence from the application tools and the wrappers and would minimize the cost for maintaining the system.* Data independence allows one to change the database design and contents while minimizing the effect to the application tools and wrappers.

Integrating a DBMS within the REF will enhance its capabilities, reduce its maintenance cost, and increase its robustness and growth potential. Areas within the REF that can take advantage of a full DBMS are shown in Figure D7, which contains the same functional blocks as the conceptual diagram shown in Figure D4. The shaded portions indicate those areas where modifications to the REF can be performed. A portion of this integration process will be re-hosting pieces of REF on a DBMS and using commercial software tools to help integrate databases. The Control Panel can be updated allowing the user access to forms for user-friendly building of queries and reports from the DBMS. These forms would add to the current capability for executing jobs within the REF. The Data Dictionary Support Software and Discipline Specific Data Dictionary functions can utilize the DBMS's imbedded data dictionary capability, e.g., its software algorithms for defining data, setting priorities, defining key words, access control, and integrating the different data definitions within domains and between domains. Database APIs are those tools that allow for report generation and query support for the casual user and for the domain specific database administrator. The Framework Administration Tools help in maintaining data integrity and concurrent engineering functions required by the different domains. Some tools within the chosen DBMS can replace current REF tools and/or work in concert with them and add additional functionality.

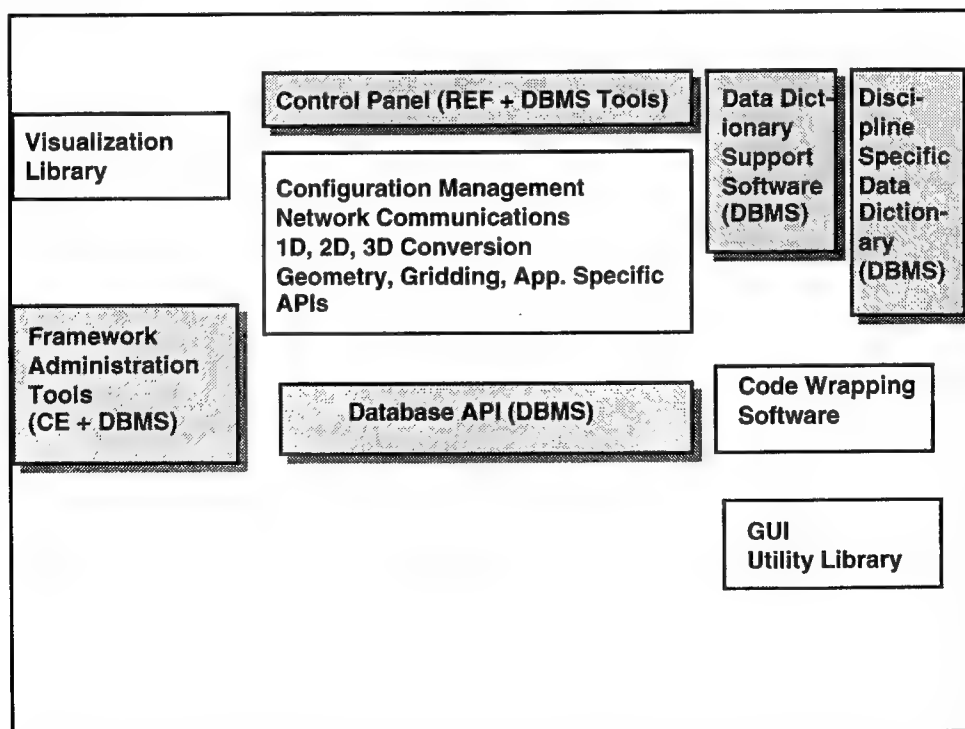


Figure D7. Re-Hosting REF Elements on a DBMS

#### **4. An Integrating REF Structure**

The previous section provided an overview and proposed a DBMS extension to the REF software architecture. This extended REF will allow it to be the foundation for integrating data from other domains. This section will describe the process of how this can be accomplished.

Within the REF the different CAD tools generate IGES files which are translated to a well-defined and common format. The non-geometry data and the Geometry API are mapped into a REF database that is FORTRAN and C compatible. The process of determining which attributes the codes share and how to describe them to build a common database definition requires domain-knowledgeable and database-knowledgeable people. The REF design and implementation process can be used as a foundation for a "bottom up" building of an integrated tool set and database for each technology domain. For example, the process that was developed for the vacuum electronics industry can be applied to the EMC domain. The process and the framework tools would be the same; but the individual translators, the data model, some of the utilities, and the database schemas would be different.

Consider the first step in applying the REF development process for the vacuum electronics industry and for the EMC community. Step one is to integrate the data from the different codes into a consistent relational DBMS (RDBMS). This will require evaluating the different codes within both technologies and defining their integrated domain databases. Once completed it will provide two of the databases as shown in Figure D2 and in Figure D8.

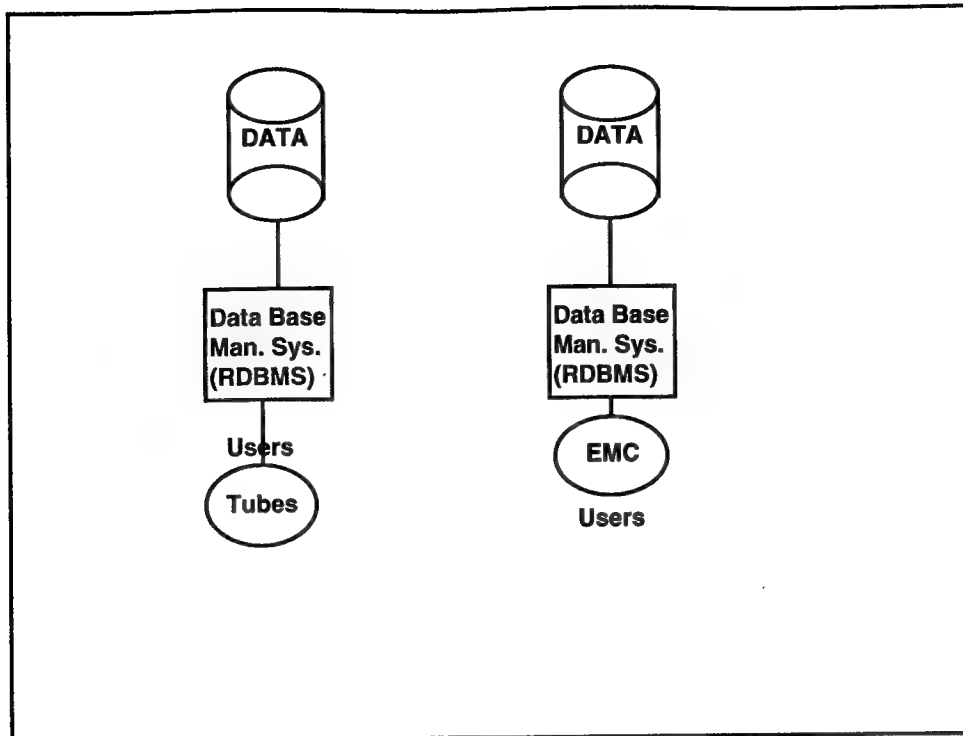


Figure D8. Building Two of the Integrated Databases

The building of each of these integrated databases can be accomplished by using the REF structure as shown in Figure D9. The vacuum electronics people are using CAD tools for their design of components, and the REF can read their output files and integrate them into a standard file system that will eventually load them into a RDBMS. Because the IGES specification is very rich in its ability, there are numerous ways for one to describe the same real world entity. This generality requires a translator that will map the different CAD tools' output files to a standard IGES file adopted by the vacuum electronics community. In this manner they have allowed for the generality and acceptance of input data from design and analysis tools and the specificity required by the database portion and the concurrent engineering community. It is the REF's user interface software and IGES translators that can be used for other domains.

The next step is to map these IGES files and parametric data to a RDBMS schema. Some of these tools have been developed within the REF, some will have to be built, and some can be obtained within the commercial community. Once the data are loaded within the RDBMS, then the user will be able to access the database, view the data, generate reports, perform queries, etc., in a consistent and unified manner. A RDBMS inherently provides a degree of data independence, security, consistency, and integrity. Most RDBMSs also provide numerous tools to easily maintain the database,

upgrade its schemas, and provide and retrieve data from software applications. In addition, by having the data within a RDBMS it allows for the eventual and easy integration of the data within a Global DBMS. Most RDBMSs are SQL compliant thereby providing for an open and easy sharing of their data across computers and RDBMSs. This will reduce the time and cost to integrate different databases.

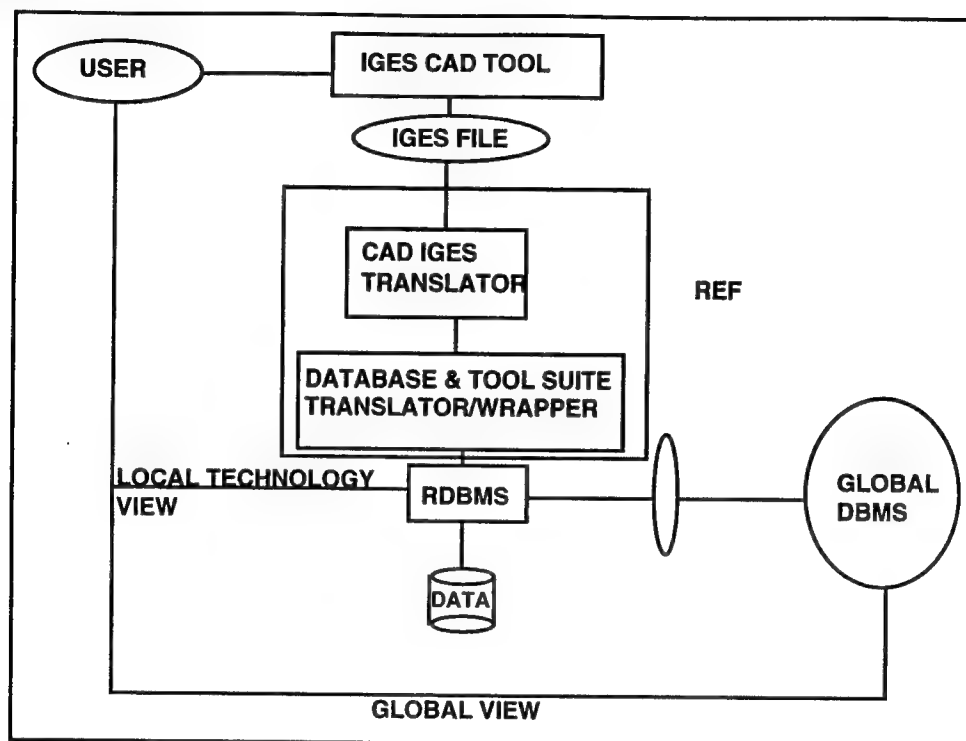


Figure D9. The REF Structure and Vacuum Electronics Integrated Databases

A similar architecture can be developed for the EMC community by replicating the development process used by the vacuum electronics industry and by utilizing a large majority of their developed software. The user interface software, IGES standard format tools, CAD IGES translator software, and database tool suite for translators and wrappers can be used and/or modified to meet the EMC environment's specifications. The first step is to define a homogeneous database from a collection of heterogeneous codes with varying data attributes, parameters, fields, names, etc. This process is labor-intensive and requires both domain-knowledgeable and database-knowledgeable people. The resultant effort will create a unified data dictionary definition of all the data attributes used for the EMC community. Through this process the requirements for the translators will also be defined. The resultant Venn diagram will be similar to the one for the vacuum electronics community (See Figure D6.) and shown in Figure D10.

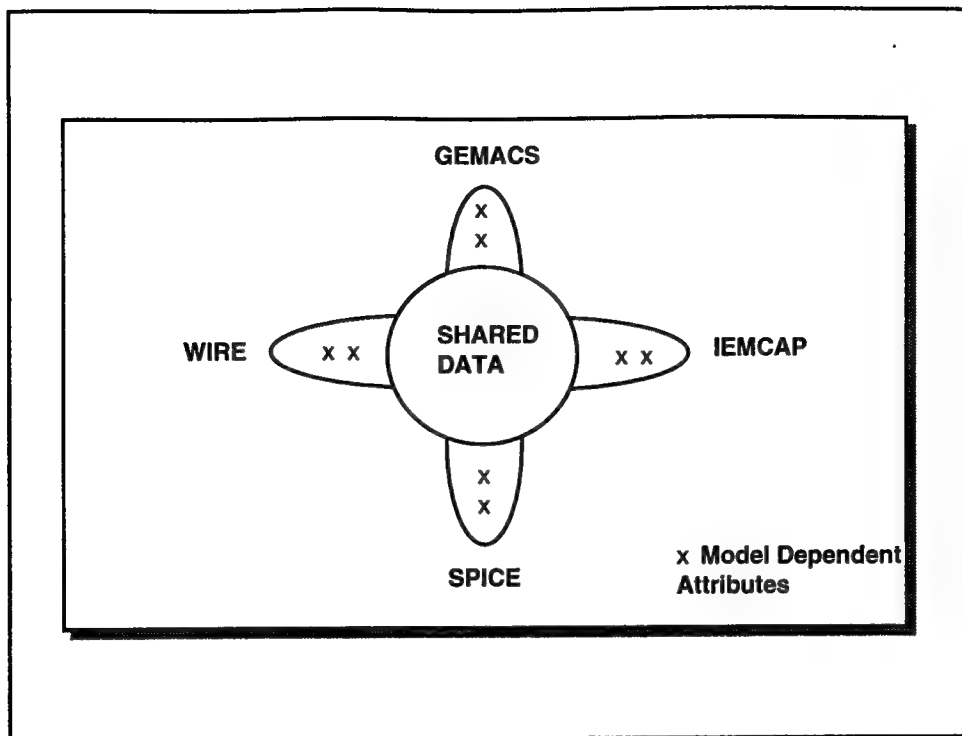


Figure D10. EMC Code Relationships

Once the data requirements for all of the EMC codes (e.g., GEMACS, IEMCAP, WIRE) are developed, then the first portion of building a consistent data dictionary for the RDBMS will have been accomplished. The next steps will be to develop an entity relationship model for the use of the data and to complete building the data dictionary and schemas. These steps are followed by building the interface tools to read the input and output files of these codes and convert them to the definitions of the database data dictionary. Some of the tools written for the REF can be used along with commercial tools to perform these functions. These tools, translators and wrappers will help provide the consistent databases required for the RDBMS. A comparison of figures D9 and D11 shows the similarities between the vacuum electronics and EMC communities when a DBMS-enhanced REF is employed in the design and analysis process.



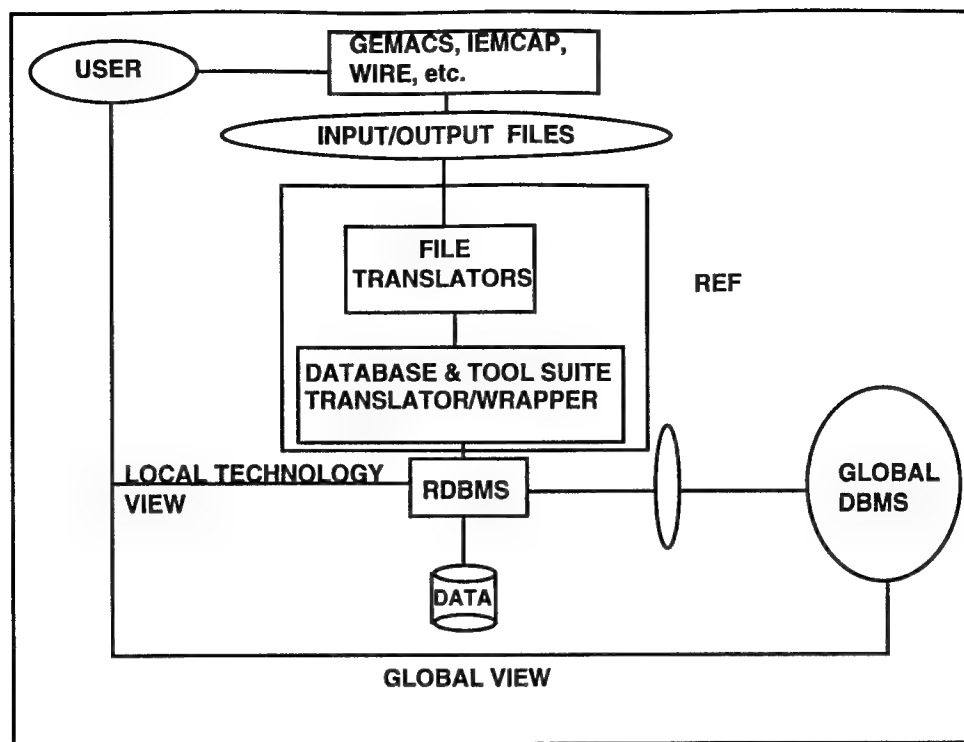


Figure D11. The REF Structure and EMC Integrated Databases

There are numerous issues that need to be addressed when integrating the data required for input and output for different codes related to the same technology domain (see Figure D11). Consider the names given to different real world entities, e.g., "bare-lead" and "pigtail". They both refer to the unshielded portion of an electrical wire, i.e., they are synonyms. There are also homonyms. The word "wire" in GEMACS refers to an element of a non-existent wire mesh model created to represent the electrical properties of a physical structure. In IEMCAP a "wire" represents an existing entity that is carrying electrical current or signals between two or more ports. There are also differences in the format of data, for example, the number of bytes set aside for each input or output field, the coding format, i.e., integer, floating point, double precision, text, and the order and/or position of the field's value when stored in a file. In addition, there are subtle coding differences that are generated among different codes. For example, dimensions are stored in inches in one code and in feet, or meters, or centimeters in another. There are differences in coding techniques for any number of fields, dates, names, and binary variables. For example, in one code "true" is represented as a +1 and "false" as a 0; in another code it may be +1 and -1 respectively. The process of integrating heterogeneous databases is a labor-intensive process.

The integration of code data is of primary importance to ICE. It is the consistent and accurate representation of the entity under investigation that is of concern. The input/output files shown coming to/from the above codes represent the entity description data (or input data to the individual codes) and the chosen output (or analysis data). These input and output data have merit as input to other codes or for comparison with the results from other codes.

*The File Translators* are those codes that understand the format of the data for each of the codes and are able to select and convert each data field that has been chosen to enter into the integrated database. They are capable of converting those selected fields within each code to an intermediate standard format that can then be integrated within the database of choice. For the IGES-generated codes the translators map the different representations to a uniform IGES representation.

For the EMC domain this requirement also exists along with mapping other inconsistencies among codes. For example, the representation of the outer structure that is modeled by different CEM codes requires that their structure representation be described in a common format in order to be represented in a consistent manner within the DBMS. Therefore, each structural representation will begin with a uniform standard. A similar type of mapping will occur for the WIRE code and IEMCAP, where wire representations and their computer description will need to be consistent before they are mapped into a DBMS. One can think of the "File Translators" as domain-specific software that converts data which represents the same world entity to a common format.

*The Database & Tool Suite Translator/Wrapper* are tools to help build the data dictionary and directory for the integrated database. The term wrapper is used because it "wraps" the code in software and performs the transfer function or the data translation to and from the different databases. These are the tools that, for example, will convert the inches to centimeters, help resolve the issues as to which attributes are synonyms and homonyms, help resolve the binary variable representation, convert integers to floating point formats and load the files in the database. These tools will also help design the integrated database system and help manage the database and its meta data. Once the data are made compatible and loaded into the RDBMS, then users can obtain access to the data via the RDBMS directly. They can then perform general queries, generate reports, maintain different code representations of the entity under study as a local technology user, and they can access the Global DBMS as a Global user.

This same procedure would be applied in developing each of the domains discussed above, i.e., EMC, vacuum tube industry. This

provides the different domains with a consistent set of data within their own DBMS.

## 5.0 Integration of Multiple Domain-Specific Relational DBMSs

The previous two sections describe how the vacuum electronics industry's REF development tools and processes can be used as a model for integrating heterogeneous databases within the vacuum electronics and the EMC domains. The REF and the process described above can also be used to integrate these two different technologies, along with numerous others, as shown in Figure D12.

To integrate the vacuum electronics database and the EMC-generated database is a matter of integrating two databases with well defined schemas and data dictionaries. Since both are assumed to be built with RDBMS SQL-compliant systems, their integration should be relatively straightforward. Data definitions, synonyms, homonyms, formats and subtle data coding differences will need to be determined and repaired based upon data and meta data intersections between the two databases. The resultant solutions will be incorporated with the global transfer functions and wrappers in a manner similar to what was done for the integration of data within each of the technology domains.

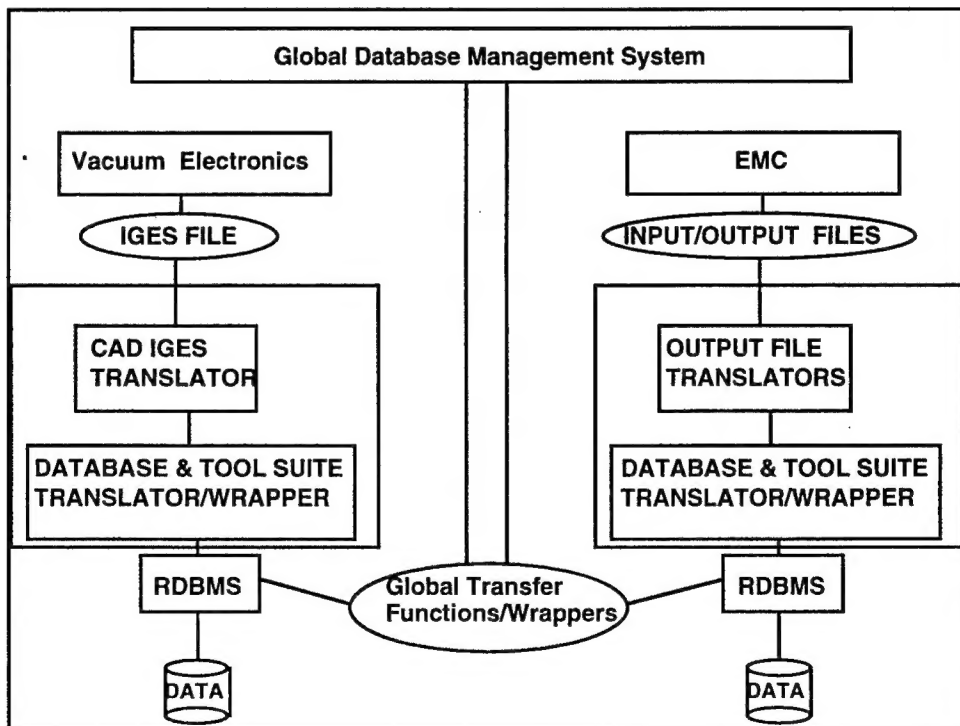


Figure D12. The REF Structure and the Integration of Two Different Database Domains

Once a global database which meets the needs of the different domain users and their views is defined, then its global data dictionary will be ready to accept global users and their views. The enhancement of the data dictionary to meet the global users' requirements will entail meetings with the users to understand the data requests they wish to make, the reports they would like to have generated, etc. These users are new users to the architecture. Initially, they are not offering to place data into the databases but wish to retrieve data from the database. As time progresses however, they will add new data to the database and/or they may add summarized data that are functions of the data within the database. For instance, they may add fields to the database that are functions of data retrieved from the database, such as statistical terms (averages, estimated variances, histograms, etc.). This process will add new fields to the data dictionary. Such data dictionary definition and the individual data dictionary definitions from each of the domains provide the basis for building the transfer functions and/or wrappers that will interface the different databases and the global users' needs. The transfer functions allow for mapping the individual database fields from a domain database to the global definitions and from a global database field to a domain database field. One can view these two-way functions and wrappers as "translators". As the architecture of integrated databases is used, the data dictionary and schemas will change to meet the continuous data additions, deletions, and needs of the multiple users.

## **6.0 Summary**

This report provided an overview of the ICE concept while adding more detail than what was provided in the first interim report. It expanded upon the architecture previously proposed where the REF was used along with DBMS technology as the host for adding technologies into an ICE architecture. The report concluded by describing how this approach for building ICE can be used to add two or more technologies to the ICE.

It is recommended that this approach be considered further as the design for ICE. In the next phase of this effort it is proposed that effort be expended to further define this design and develop a plan for the building of ICE.

## References

- \*1. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Phase II Basic Final Report," CDRL A003, Draft A, March 1995.
- \*2. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Reference & Specification Document," CDRL A005, Draft A, March 1995.
- \*3. Hantman, B., Hicks, R., and LaBelle, J., "Microwave/Millimeter-wave Advanced Computational Environment (MMACE) Research & Engineering Framework (REF) Phase II Basic Design Tool Integration Document," CDRL A006, Draft A, March 1995.
- 4. Capraro, Gerard, T., "Integrated Computational Environment (ICE) First Interim Report," October 1995.

\*Items 1,2, & 3 are Disbribution Authorized to DOD Components Only.

***MISSION  
OF  
ROME LABORATORY***

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

- a. Conducts vigorous research, development and test programs in all applicable technologies;
- b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;
- c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;
- d. Promotes transfer of technology to the private sector;
- e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.